

Handling Constrained Many-objective Optimization Problems via Problem Transformation

Ruwang Jiao, Sanyou Zeng, Changhe Li, *Member, IEEE*,
Shengxiang Yang, *Senior Member, IEEE*, and Yew-Soon Ong, *Fellow, IEEE*

Abstract—Objectives optimization and constraints satisfaction are two equally important goals to solve constrained many-objective optimization problems (CMAOPs). However, most existing studies for CMAOPs can be classified as feasibility-driven constrained many-objective evolutionary algorithms (C-MaOEAs), they always give priority to satisfy constraints, while ignoring the maintenance of the population diversity for dealing with conflicting objectives. Consequently, the population may be pushed towards some locally feasible optimal or locally infeasible areas in the high-dimensional objective space. To alleviate this issue, this paper presents a problem transformation technique, which transforms a CMAOP into a dynamic CMAOP (DCMAOP) for handling constraints and optimizing objectives simultaneously, to help the population cross the large and discrete infeasible regions. The well-known reference-point-based NSGA-III is tailored under the problem transformation model to solve CMAOPs, namely DCNSGA-III. In this paper, ϵ -feasible solutions play an important role in the proposed algorithm. To this end, in DCNSGA-III, a mating selection mechanism and an environmental selection operator are designed to generate and choose high-quality ϵ -feasible offspring solutions, respectively. The proposed algorithm is evaluated on a series of benchmark CMAOPs with 3, 5, 8, 10, and 15 objectives and compared against six state-of-the-art CMAOEAs. The experimental results indicate that the proposed algorithm is highly competitive for solving CMAOPs.

Index Terms—Evolutionary computation, constrained optimization, many-objective optimization, problem transformation

I. INTRODUCTION

A large number of optimization problems arise in the science and engineering areas, where there is a need to balance multiple conflicting objectives without violating constraints. They are known as constrained multi-objective

optimization problems (CMOPs). Particularly, problems with four or more objectives are named as constrained many-objective optimization problems (CMAOPs). A CMAOP can be mathematically formulated as the minimization of m objective functions subject to a set of constraints:

$$\begin{aligned} \text{minimize} \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})) \\ \text{subject to:} \quad & \mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), \dots, g_q(\mathbf{x})) \leq \mathbf{0} \\ \text{where} \quad & \mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathbf{X} \\ & \mathbf{X} = \{\mathbf{x} | \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}\} \\ & \mathbf{l} = (l_1, l_2, \dots, l_n), \mathbf{u} = (u_1, u_2, \dots, u_n), \end{aligned} \quad (1)$$

where \mathbf{x} is the solution vector of which x_k is within $l_k \leq x_k \leq u_k$ ($k = 1, \dots, n$), $\mathbf{f}(\mathbf{x})$ is the objective vector that consists of m real-valued objective functions, $\mathbf{g}(\mathbf{x})$ is the vector of inequality constraints, $\mathbf{0}$ denotes the constraint boundary. If a solution \mathbf{x} satisfies $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$, it is called a feasible solution; otherwise, it is an infeasible solution.

Optimization problems with a high number of objectives appear widely in real-world applications [1]–[5]. The recent ten years have witnessed significant progress in the development of many-objective evolutionary algorithms (MaOEAs) for solving many-objective optimization problems (MaOPs) to overcome the difficulty of the weak selection pressure. The algorithms can be categorized into seven classes [6]: relaxed dominance-based, diversity-based, aggregation-based, indicator-based, preference-based, reference set based, and dimensionality reduction approaches. Although many MaOEAs have been proposed during the past decade, the majority of them are for unconstrained MaOPs, there are only a few works dealing with CMAOPs. CMAOPs frequently occur in many scientific and engineering domains, most practical optimization problems cannot be formulated without constraints [7]. Compared with unconstrained MaOPs, CMAOPs are much more difficult since constraints make feasible regions discrete. Individuals of a population are hard to get across the infeasible regions to find the optimal feasible solutions in the high-dimensional objective space.

When solving CMAOPs, two important issues are how to handle constraints and how to balance convergence and diversity in the high-dimensional objective space. Nevertheless, most CMAOEAs overly emphasize the significance of feasibility [8], they tend to push a population towards feasible regions, whereas they rarely consider the balance among convergence, diversity, and feasibility simultaneously. This may result in the population being stuck at some locally optimal or locally feasible regions, especially when feasible regions are narrow or disjointed distributed in the search space [8], [9]. To alleviate

This work was supported in part by the National Natural Science Foundation of China under Grant 62076226, 61673355 and Grant 61673331, in part by the Hubei Provincial Natural Science Foundation of China under Grant 2015CFA010, in part by the 111 project under Grant B17040, in part by the Fundamental Research Funds for National Universities, China University of Geosciences(Wuhan) under Grant CUGGC02. (Corresponding authors: Sanyou Zeng; Changhe Li)

R. Jiao and S. Zeng are with the School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan, China, 430074. (e-mail: ruwangjiao@gmail.com, sanyouzeng@gmail.com).

C. Li is with the School of Automation, China University of Geosciences, Wuhan, China, 430074, and also with the Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, Wuhan 430074, China (e-mail: changhe.li@gmail.com).

S. Yang is with the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, U.K. (e-mail: syang@dmu.ac.uk).

Y.-S. Ong is with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: asysong@ntu.edu.sg).

the above issue, this paper proposes a problem transformation technique to solve CMaOPs. The main contributions of this paper are highlighted as follows:

- 1) A problem transformation technique is proposed. It transforms a MaOP with highly-constraints into a DC-MaOP, where the constraints are easy to satisfy. Under the proposed problem transformation technique, the constraint difficulty is increased little by little as the search goes on. For a specific moment, the transformed problem can be solved by a CMaOEA in a way of solving an unconstrained MaOP. The dynamic constraints can help a MaOEA to focus on the tradeoff between diversity and convergence.
- 2) A DCNSGA-III algorithm is developed under the proposed problem transformation model. The key to the success of the proposed algorithm is to maintain a high ratio of ε -feasible solutions in the population. To achieve this goal, in DCNSGA-III, a modified tournament selection operator is put forward to generate more ε -feasible offspring solutions. Furthermore, an environment selection process combined with the ε -constrained dominance principle is designed to identify potential solutions by considering diversity, convergence, and constraints. Compared with the original NSGA-III [10], the proposed DCNSGA-III does not introduce any complex operator, and there is no extra computational burden for computational complexity.
- 3) A comprehensive comparison is performed for a number of state-of-the-art CMaOEAs on existing CMOPs with 3, 5, 8, 10, and 15 objectives, including CFs [11], C-DTLZs [12], DC-DTLZs [8], MWs [13], and DASCMPs [14]. The experimental results show that DCNSGA-III can obtain competitive results on most benchmark test suites.

The rest of this paper is organized as follows. Section II reviews the related work on constraint-handling techniques, and CMaOEAs. Section III presents the problem transformation technique in detail. Section IV provides an instantiation called DCNSGA-III. Experimental results and comparison with six representative algorithms are shown in Section V. A further investigation of the performance analysis is conducted in Section VI. Finally, Section VII draws conclusions and identifies directions of future work.

II. RELATED WORK

A. Introduction to Constraint-handling techniques

Roughly speaking, constraint-handling techniques based on evolutionary algorithms (EAs) for CMOPs can be briefly classified into four categories [15]: penalty function, separation of objectives and constraints, multi-objective methods and ensemble of constraint-handling techniques.

The penalty function method is one of the simplest and oldest constraint-handling techniques. It introduces a penalty coefficient, which is multiplied with the constraint violation into the original objective function to penalize those solutions that violate constraints. There are three ways to set the value of

the penalty coefficient: static, dynamic, and adaptive penalty coefficient.

Constraint dominance principle (CDP) [16] is an extension of dominance principle for solving CMOPs, which does not require fine-tuning parameters. For any two solutions, \mathbf{x}_1 is better than \mathbf{x}_2 under the following three conditions:

- \mathbf{x}_1 is feasible while \mathbf{x}_2 is not;
- Both of them are infeasible, but the degree of constraint violation of \mathbf{x}_1 is smaller than \mathbf{x}_2 ;
- Both of them are feasible, but \mathbf{x}_1 Pareto dominates \mathbf{x}_2 .

The strong selection pressure of CDP can result in the fast convergence towards the feasible area, which is easy to lose the diversity for a population. The angle information between every two solutions can reflect the degree of diversity. Using angle information between solutions in the objective space to maintain a more fine-grained diversity has been investigated in solving MaOPs [17], [18], and also extended to the CDP to handle constraints [19]. For example, angle-based CDP [19] employs the angle information among solutions and the feasible ratio to adjust the dominance relationship, so that it can maintain the proper diversity when handling constraints.

The optimum of most constrained optimization problems (COPs) lies on the boundary of the feasible region, it is beneficial to utilize not only feasible solutions but also infeasible solutions [20]. Identifying and maintaining promising infeasible solutions close to the feasible region can improve the efficiency of the optimization process, so that more solutions will be generated inside the feasible region and also near its boundaries by using genetic operators [21]. In addition, the presence of infeasible solutions might enable the search to move between disjoint feasible regions, to more-easily explore solutions at the edge of feasible regions, or to take a shorter path to the global optimum by traversing the infeasible part of the search space [22]. Various forms of preference articulation schemes of infeasible solutions have been used in various stages of the solution process, e.g., problem formulation, parent selection/recombination and ranking/selection schemes [23].

The stochastic ranking method [24] and the ε constraint method [25] are the two most used methods using infeasible solutions, which introduce the objective information during the comparison process. In the stochastic ranking method, the comparison between two adjacent solutions based on either objective values or the constraint violation is dependent on a probability, which indicates infeasible solutions have a chance of being preserved. In the ε constraint method, the comparison criterion between two solutions switches between the case where objective functions precede constraint violation ($\varepsilon = \infty$) and the case where constraint violation precedes objective functions ($\varepsilon = 0$) according to the ε level. The ε constraint method allows for the preservation of marginally infeasible solutions.

The multi-objective optimization method is a promising technique because it can provide a balance between objectives and constraints [26]. By regarding the constraint violation as an objective or each constraint as an objective, a CMOP is converted to an $(m + 1)$ -objective optimization problem or

an $(m + q)$ -objective optimization problem, where q is the constraint number.

Recently, the dynamic multi-objective technique has shown promising performance for solving constrained single-objective optimization problems [27], [28]. It converts a COP to a dynamic constrained MOP which includes three objectives: the original objective, the constraint violation objective, and the niche-count objective. The constraint violation objective is used to handle the constraint difficulty and the niche-count objective is utilized to maintain the population diversity in the decision space.

B. Related Work on CMAOEs

For most existing CMAOEs, feasibility takes precedence over diversity and convergence [12], [18], [29], where they normally integrate the well-known CDP [16] to select elite solutions with the preference of feasible solutions, due to the appealing advantages of CDP: it is easy to be implemented and free of setting of parameters, it is also capable of driving the population to converge to the feasible area quickly. The most representative algorithm is C-NSGA-III [12], which is a combination of CDP and NSGA-III [10]. NSGA-III is an improved version of NSGA-II [16], which tries to relieve the severe loss of Pareto-based selection pressure toward the PF [30]. It replaces the crowding distance operator in NSGA-II with a clustering operator for clustering a set of well-distributed reference points, which are used to generate niches, associate and select elite solutions.

Convergence, diversity, and feasibility can be seen as three indicators for CMAOPs. TiGE [31] and C-TAEA [8] were developed based on these three indicators. In TiGE, a variety of balance schemes and ranking methods can be used to achieve the balance among these three indicators. C-TAEA maintains two collaborative archives: a convergence-oriented archive focuses on maintaining the convergence and feasibility of the evolution process, a diversity-oriented archive tends to explore the feasible and infeasible regions to provide more diversified information.

Leveraging the information of infeasible solutions can improve the population diversity. C-MOEA/DD [32] reserves infeasible solutions when they are associated with isolated subregions for escaping from locally feasible areas. Nevertheless, feasible solutions are still favored over infeasible solutions in the mating selection operator. In C-RVEA [33], the uniformly distributed reference vectors are adopted to cluster the population and then select a solution in each cluster. Although feasible solutions are still favored over infeasible solutions in each cluster, the clustering operation could preserve an infeasible solution in a sparse area. A detect-and-escape strategy [34] is employed to detect whether the search is stuck in a feasible subregion or an infeasible local region. To be specific, it adopts the feasible ratio and the change rate of the degree of constraint violation to detect stagnation of an algorithm, and then adjusts the weight of the constraint violation accordingly for guiding the search to escape from stagnation states.

Infeasible solutions with good convergence could help the population to get cross the infeasible barrier. Push and pull

search (PPS) [35] divides the search process into two phases: in the push phase, the population is pushed to cross infeasible regions in front of the unconstrained PF by ignoring constraints. In the pull phase, the population is pulled to the feasible and non-dominated region by employing an enhanced ϵ constraint method. In I-DBEA [36], the ϵ level for distinguishing the feasible and infeasible solutions is adaptively adjusted based on the average level of constraint violation and the feasibility ratio of the population.

C. Discussions

Both feasible and infeasible solutions are important for solving CMAOPs. Only preferring feasible solutions over infeasible solutions could lead to two consequences:

- 1) Searching only guided by constraint violation may make the population stagnate in the infeasible region easily, or make the population gather in a small feasible region, which may result in premature convergence;
- 2) Due to the high selection pressure for feasible solutions, the diversity of the population will drastically decrease. If the feasible region consists of several disjoint parts, the population may miss some of them, resulting in the missing of some Pareto optimal solutions.

Nevertheless, not all infeasible solutions are beneficial to the search:

- 1) Allowing infeasible solutions to survive during the course of selection will enlarge the search space and may result in wasting resources for an algorithm running up blind alleys;
- 2) Using misleading infeasible solutions will lead to the search trapping in infeasible local optima and be unable to return to the feasible part of the search space.

Apparently, in different search stages, the use of infeasible solutions should be different. In the next section, we would like to provide a problem transformation technique, which can take the constraint satisfaction and the balance between diversity and convergence into account simultaneously. In the early stage of the optimization, the proposed method utilizes a large number of promising infeasible solutions to force the population to towards the unconstrained PF. In the middle and later stages of the optimization, the search switches from optimality priority to feasibility priority, which means it searches Pareto-optimal solutions mainly from the feasible space.

Note the proposed method is based on our previous work [27], [28], they all adopt dynamic constrained multi-objective evolutionary algorithms to handle constraints. In [27], [28], a COP was transformed into an equivalent dynamic bi- or tri-objective optimization problem. However, compared with [27], [28], this paper mainly has three differences:

- 1) This paper focuses on addressing CMAOPs. Different from COPs to find a global optimum, CMAOPs are more complicated due to they need to consider the diversity, convergence, and constraints simultaneously.
- 2) This paper proposes a mating selection strategy to generate more high-quality offsprings. In the high-dimensional objective space, the choice of parents from

the entire population is not a good idea since many inefficient offsprings will be created [37], [38]. To improve effects of the genetic operation, the proposed mating selection strategy can produce more efficient ε -feasible solutions.

- 3) This paper proposes an environment selection scheme to select promising ε -feasible solutions. This scheme does not need to set additional reference points for the constraint violation objective.

III. PROBLEM TRANSFORMATION TECHNIQUE

A. Motivation

As described in Section I, there are two difficulties for solving CMaOPs: how to balance the diversity and convergence, and how to tackle constraints efficiently in the high-dimensional objective space. For the first difficulty, the development of unconstrained MaOEAs can be adopted. For the second difficulty, although several constraint-handling techniques have been proposed, most of them are designed for single-objective COPs, only a few works have been done in handling CMaOPs due to constraints making feasible areas separate in the high-dimensional objective space, it is more difficult to push the population to cross the multiple local feasible and infeasible regions to approach the constrained PF. To alleviate this issue, we transform a CMaOP into a DCMaOP by regarding the constraint violation as an extra objective. In addition, the constraint boundary is relaxed to make all individuals ε -feasible at the beginning of the run, and it is shrunk dynamically as the search goes on. In this way, infeasible solutions with good convergence and diversity, as well as with smaller constraint violation values, can be consistently reserved, with the purpose of getting through the large infeasible barrier to escape from the locally feasible or infeasible regions. Meanwhile, by transforming a CMaOP into a DCMaOP, at each generation, the population focus on searching for the PF under the current generation, which could maintain the completeness of the final PF.

B. Transformation of a CMaOP into a DCMaOP

A DCMaOEA adopts the constraint-violation as an added objective to address CMaOPs. It recasts an m -objective C-MaOP $(f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$ as an $(m+1)$ -objective CMaOP $(f_1(\mathbf{x}), \dots, f_m(\mathbf{x}), cv(\mathbf{x}))$. The constraint violation is a widely-used measure for dealing with COPs. In this paper, the normalized average degree of the constraint violation of \mathbf{x} on all constraints is regarded as the constraint violation objective:

$$cv(\mathbf{x}) = \frac{1}{q} \sum_{i=1}^q \frac{G_i(\mathbf{x})}{\max_{\mathbf{x} \in \mathcal{P}_0} \{G_i(\mathbf{x})\}}, \quad (2)$$

where \mathcal{P}_0 is the initial population, $G_i(\mathbf{x})$ is the degree of the constraint violation of \mathbf{x} on the i th constraint:

$$G_i(\mathbf{x}) = \max\{g_i(\mathbf{x}), 0\}, i = 1, 2, \dots, q. \quad (3)$$

Note that when $\max_{\mathbf{x} \in \mathcal{P}(0)} \{G_i(\mathbf{x})\} < 1, i = 1, 2, \dots, q$, we replace the $\max_{\mathbf{x} \in \mathcal{P}(0)} \{G_i(\mathbf{x})\}$ with 1.

We first construct a DCMaOP:

$$\begin{aligned} & \text{minimize} && (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}), cv(\mathbf{x})) \\ & \text{subject to :} && \mathbf{g}(\mathbf{x}) \leq \boldsymbol{\varepsilon}^{(t)} \\ & \text{where} && \boldsymbol{\varepsilon}^{(t)} = (\varepsilon_1^{(t)}, \dots, \varepsilon_q^{(t)}), t = 0, 1, \dots, T \end{aligned} \quad (4)$$

where $\boldsymbol{\varepsilon}^{(t)}$ is the dynamic constraint boundary, $t = 0, 1, \dots, T$. T is the maximum number of environmental changes. A solution, which satisfies $\mathbf{g}(\mathbf{x}) \leq \boldsymbol{\varepsilon}^{(t)}$, is called an ε -feasible solution; otherwise, it is called an ε -infeasible solution.

The problem formulation of Eq. (4) is different from the original CMaOP problem formulation in Eq. (1) with two changes: (1) the constraint violation objective is added; (2) the dynamic constraint boundary $\boldsymbol{\varepsilon}$ is enlarged. To build an equivalent problem transformation, the $\boldsymbol{\varepsilon}$ constraint boundary is dynamically reduced along with the environment state: $\boldsymbol{\varepsilon}^{(0)} > \boldsymbol{\varepsilon}^{(1)} > \dots > \boldsymbol{\varepsilon}^{(T)} = \mathbf{0}$. An environmental change from state t to $t+1$ represents the shrink of the constraint boundary. At the last state T , we set $\boldsymbol{\varepsilon}^{(T)} = \mathbf{0}$, $cv(\mathbf{x}) = 0$, which means Eq. (4) has the following simplified form:

$$\begin{aligned} & \text{minimize} && (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}), 0) \\ & \text{subject to :} && \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \end{aligned} \quad (5)$$

note that at the last state T , an ε -feasible solution who satisfies $\mathbf{g}(\mathbf{x}) \leq \boldsymbol{\varepsilon}^{(T)}$ is actually a feasible solution of the original problem (1) at this final state T , it is also called a feasible solution since $\boldsymbol{\varepsilon}^{(T)} = \mathbf{0}$.

From problem formulation (4), we can see that with the environment stage changes from state t to state $t+1$, the problem to be solved in state $t+1$ is different from the problem in state t . This is because with the $\boldsymbol{\varepsilon}$ constraint boundary shrinks from $\boldsymbol{\varepsilon}^{(t)}$ to $\boldsymbol{\varepsilon}^{(t+1)}$, the Pareto-optimal solutions from state t to state $t+1$ are also changed. The algorithm can concentrate on searching Pareto-optimal solutions at state t on condition that the majority solutions of the population are ε feasible ($\mathbf{g}(\mathbf{x}) \leq \boldsymbol{\varepsilon}^{(t)}$). In the next section, we will discuss how to generate and keep more ε -feasible solutions.

In this paper, **ε -constrained dominance principle (ε -CDP)** is employed to compare a pair of solutions. For any two solutions, \mathbf{x}_1 is said to ε -constrained Pareto dominates \mathbf{x}_2 , if

- \mathbf{x}_1 is ε -feasible and \mathbf{x}_2 is ε -infeasible;
- \mathbf{x}_1 and \mathbf{x}_2 are both ε -infeasible, but the constraint violation of \mathbf{x}_1 is smaller than \mathbf{x}_2 ;
- \mathbf{x}_1 and \mathbf{x}_2 are both ε -feasible, but \mathbf{x}_1 dominates \mathbf{x}_2 based on $(f_1(\mathbf{x}), \dots, f_m(\mathbf{x}), cv(\mathbf{x}))$.

Formally, the transformed problem looks more complex as it adds one more objective. In fact, the transformation makes the problem easier to solve than before because the constraint difficulty has been removed through the transformation.

Fig. 1 shows the working mechanism of the proposed problem transformation technique, where the gray areas are infeasible regions, the white areas in the attainable objective space represent feasible regions, and the green areas are ε -feasible regions. As shown in Fig. 1(a), initial solutions were generated and they are all located in infeasible regions away from the PF. Infeasible regions block the way of the population towards the PF, it is hard for the population

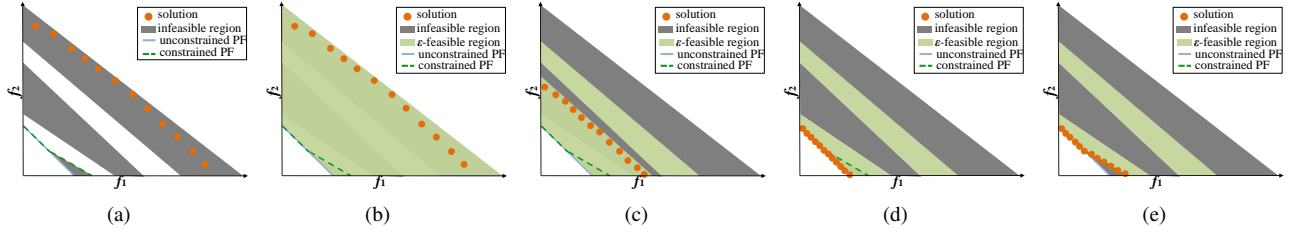


Fig. 1. The illustration of how the population crosses infeasible regions during the search. (a) An initial population is generated, all solutions are infeasible; (b) the original feasible region is relaxed to the relatively large ϵ -feasible region, so that all solutions of the initial population are ϵ feasible; (c) in the middle stage of the evolution, the population searches in the ϵ feasible region; (d) in the late stage of the evolution, the population gets to the unconstrained PF; (e) at the last generation, the dynamic ϵ feasible region shrinks to the original feasible region, the population reaches to the constrained PF.

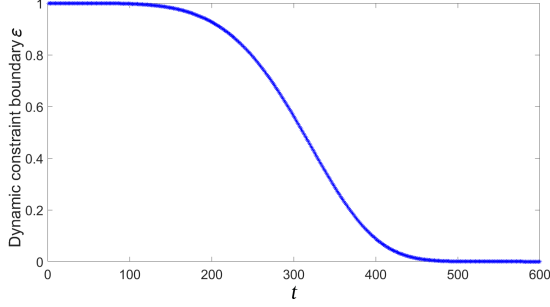


Fig. 2. Dynamic changing ϵ constraint boundary with $\epsilon^{(0)} = 1$ and $T=600$.

to get across obstacles caused by the two large infeasible regions. However, as illustrated by Fig. 1(b), our proposed method first extends the size of the feasible region by the ϵ parameter, solutions inside the ϵ constraint boundary can be conditionally considered as feasible (ϵ feasible), so the search can be performed in a relatively large space to maintain population diversity, and get across the large infeasible regions and approach the unconstrained PF even if many local optima exist in infeasible regions. When the relaxed ϵ -feasible space gradually shrunk into original feasible regions, as illustrated in Fig. 1(e), all infeasible solutions in the unconstrained PF move towards the non-dominated constraint boundary, which means the population reaches the constrained PF.

The shrink of the ϵ dynamic constraint boundary adopts the exponential function of the simulated annealing algorithm [39]:

$$\epsilon_i^{(t)} = A_i e^{-\left(\frac{t}{B_i}\right)^{cp}} - \delta, i = 1, 2, \dots, q \quad (6)$$

where δ is a close-to-zero value ($\delta = 1e-8$), cp controls the decreasing trend of ϵ .

In order to make the initial parent population \mathcal{P}_0 achieves ϵ feasible, the maximum violation of a constraint in \mathcal{P}_0 is selected as the initial constraint boundary: $\epsilon_i^{(0)} = \max_{\mathbf{x} \in \mathcal{P}_0} \{G_i(\mathbf{x})\}$, $i = 1, 2, \dots, q$ at $t = 0$ for $G_i(\mathbf{x})$. The last dynamic constraint boundary is $\epsilon^{(T)}=0$ at $t = T$. From the initial and the last states, the constants A_i and B_i can be obtained, respectively:

$$\begin{cases} A_i = \epsilon_i^{(0)} + \delta \\ B_i = \frac{T}{cp \sqrt[cp]{\ln\left(\frac{\epsilon_i^{(0)} + \delta}{\delta}\right)}} \end{cases} \quad (7)$$

Algorithm 1: General framework of DCNSGA-III for CMAOPs

- 1 Generate an initial population \mathcal{P}_0 ;
 - 2 Relax the initial constraint boundary $\epsilon = \epsilon^{(0)}$, and set the problem state $t = 0$;
 - 3 **while** the halting criterion is not satisfied **do**
 - 4 Shrink the boundary: $\epsilon = \epsilon^{(t+1)}$;
 - 5 Update population ϵ -feasibility, $t = t+1$;
 - 6 Generate the offspring population \mathcal{Q}_t according to Algorithm 2;
 - 7 Update the parent population \mathcal{P}_{t+1} ; /*Algorithm 3*/
 - 8 **end**
 - 9 return non-dominated solutions.
-

The changes of ϵ dynamic constraint boundary as the environment state increases are plotted in Fig. 2, where the initial dynamic constraint boundary and the maximum number of environmental changes are set to $\epsilon^{(0)} = 1$ and $T=600$, respectively. From Fig. 2, we can see that in the early stage of the evolution, the ϵ dynamic constraint boundary shrinks very slowly, a DCMAOEA can put more emphasis on exploration in infeasible regions without considering too much constraints; in the middle stage of the evolution, the ϵ dynamic constraint boundary drops very fast, the search switches from optimality priority to feasibility priority; in the later stage of the evolution, the ϵ dynamic constraint boundary shrinks to its original constraint boundary, which guarantees DCMAOEA searches Pareto-optimal solutions in the feasible space.

IV. THE PROPOSED ALGORITHM: DCNSGA-III

In this section, we integrate a well-known reference-point-based non-dominated sorting algorithm NSGA-III [10] with the problem transformation model, referred as DCNSGA-III, for solving CMAOPs. Algorithm 1 presents the framework of DCNSGA-III for solving CMAOPs.

DCNSGA-III begins with a randomly generated population \mathcal{P}_0 in a unified search space \mathbf{X} . The constraint violation for each solution is then calculated, and the maximum constraint violation is taken as the initial ϵ constraint boundary, so that all solutions in the population \mathcal{P}_0 are ϵ -feasible. At each iteration, the ϵ constraint boundary will be reduced as the number of generations increases.

As discussed before, ϵ -feasible solutions play a critical role in the proposed algorithm. To create high-quality ϵ -feasible

Algorithm 2: Tournament Selection Procedure

Input: candidate solutions \mathbf{x}_1 and \mathbf{x}_2
Output: mating parent \mathbf{p}

```

1 if  $\mathbf{x}_1$  is  $\varepsilon$ -feasible and  $\mathbf{x}_2$  is  $\varepsilon$ -infeasible then
2    $\mathbf{p} \leftarrow \mathbf{x}_1$ ;
3 else if  $\mathbf{x}_1$  is  $\varepsilon$ -infeasible and  $\mathbf{x}_2$  is  $\varepsilon$ -feasible then
4    $\mathbf{p} \leftarrow \mathbf{x}_2$ ;
5 else if  $\mathbf{x}_1$  is  $\varepsilon$ -infeasible and  $\mathbf{x}_2$  is  $\varepsilon$ -infeasible then
6   if  $cv(\mathbf{x}_1) < cv(\mathbf{x}_2)$  then
7      $\mathbf{p} \leftarrow \mathbf{x}_1$ ;
8   else if  $cv(\mathbf{x}_1) > cv(\mathbf{x}_2)$  then
9      $\mathbf{p} \leftarrow \mathbf{x}_2$ ;
10  else
11     $\mathbf{p} \leftarrow \text{random}(\mathbf{x}_1, \mathbf{x}_2)$ ;
12  end
13 else
14    $\mathbf{p} \leftarrow \text{random}(\mathbf{x}_1, \mathbf{x}_2)$ ;
15 end
16 return  $\mathbf{p}$ .
```

solutions, a modified mating selection and an environment selection mechanism are designed, which will be described below.

A. Select mating parents

Ideally, we hope every solution in the population at any state is within the relaxed ε constraint boundary, which signifies all solutions are ε feasible, thus a MaOEA can focus on balancing the convergence and diversity within the ε feasible regions without considering the original constraints. However, ε -infeasible solutions cannot be avoided. To generate more ε -feasible offspring solutions, we prefer an ε -feasible solution to an ε -infeasible solution, or a solution with a small constraint violation to a solution with a large constraint violation. For this purpose, in the offspring generation operator, we randomly select two individuals from \mathcal{P}_t , and use a binary tournament selection. Algorithm 2 describes the tournament selection procedure, where an ε -feasible solution is chosen over an ε -infeasible solution or a solution with a smaller constraint violation between two ε -infeasible solutions is chosen.

B. Update the parent population

Algorithm 3 gives the procedure of updating the parent population. At each generation t , the union of the parent population and the offspring population is divided into an ε -feasible set $\mathcal{S}_1 = \{\mathbf{x} \in \mathcal{U}_t | g(\mathbf{x}) \leq \varepsilon^{(t)}\}$ and an ε -infeasible set $\mathcal{S}_2 = \{\mathbf{x} \in \mathcal{U}_t | \exists g_j(\mathbf{x}) > \varepsilon_j^{(t)}, j \in \{1, \dots, q\}\}$. We prefer to select ε -feasible solutions in \mathcal{S}_1 . To be specific, two cases are under our consideration:

- If $|\mathcal{S}_1| \geq N$, the non-dominated sorting and reference-point-based elite selection are conducted to select N solutions out of \mathcal{S}_1 . These selected N solutions are added to \mathcal{P}_{t+1} . The details of this procedure will be described in Section IV-C;

Algorithm 3: Update the parent population

Input: parent population \mathcal{P}_t , offspring population \mathcal{Q}_t
Output: parent population \mathcal{P}_{t+1}

```

1  $\mathcal{U}_t = \mathcal{P}_t \cup \mathcal{Q}_t$ ;
2 Divide  $\mathcal{U}_t$  into the  $\varepsilon$ -feasible set  $\mathcal{S}_1 = \{\mathbf{x} \in \mathcal{U}_t | g(\mathbf{x}) \leq \varepsilon^{(t)}\}$  and the  $\varepsilon$ -infeasible set  $\mathcal{S}_2 = \{\mathbf{x} \in \mathcal{U}_t | \exists g_j(\mathbf{x}) > \varepsilon_j^{(t)}, j \in \{1, \dots, q\}\}$ .
3 if  $|\mathcal{S}_1| \geq N$  then
4   Implement the non-dominated sorting and reference-point-based elite selection to select  $N$  solutions out of  $\mathcal{S}_1$ ; /* Algorithm 4 */
5   Put the selected  $N$  solutions into  $\mathcal{P}_{t+1}$ ;
6 else
7   Put solutions in  $\mathcal{S}_1$  into  $\mathcal{P}_{t+1}$ ;
8   Sort solutions in  $\mathcal{S}_2$  according to their degree of constraint violation, and put the best  $(N - |\mathcal{S}_1|)$  solutions in the sorted  $\mathcal{S}_2$  into  $\mathcal{P}_{t+1}$ ;
9 end
10 return  $\mathcal{P}_{t+1}$ .
```

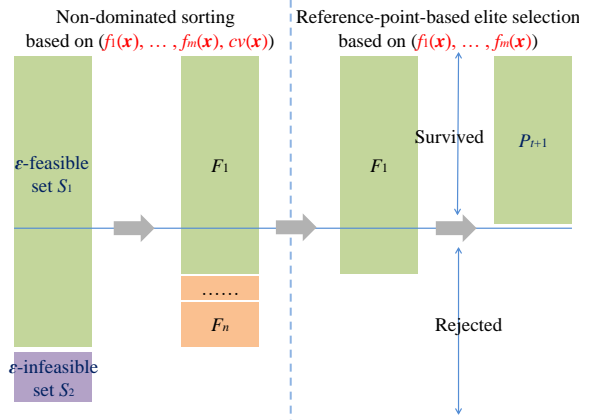


Fig. 3. Environment selection of DCNSGA-III.

- If $|\mathcal{S}_1| < N$, which means the number of ε -feasible solutions is less than the population size N , all solutions in \mathcal{S}_1 are directly added to \mathcal{P}_{t+1} . The rest $(N - |\mathcal{S}_1|)$ solutions are chosen from \mathcal{S}_2 . Herein, we sort the solutions in \mathcal{S}_2 according to the degree of their constraint violation, and then put the top $(N - |\mathcal{S}_1|)$ solutions in the sorted \mathcal{S}_2 into \mathcal{P}_{t+1} .

C. Reference-point-based non-dominated sorting

If the number of ε -feasible solutions is greater than the population size N , the reference-point-based non-dominated sorting will be conducted as the environment selection. Fig. 3 depicts the environment selection of DCNSGA-III, it consists of two steps: 1) the non-dominated sorting, which intends to choose non-dominated solutions from the ε -feasible set \mathcal{S}_1 to get close to its PF; 2) the reference-point-based elite selection, which aims at maintaining the population diversity by supplying a set of well-distributed reference points.

Firstly, ϵ -feasible set \mathcal{S}_1 is classified into different non-dominated levels (F_1, F_2, \dots) by using the non-dominated sorting procedure based on $m + 1$ objectives: $(f_1(\mathbf{x}), \dots, f_m(\mathbf{x}), cv(\mathbf{x}))$, with the purpose of selecting elite solutions with better objective values and smaller constraint violation values. By regarding the constraint violation as an additional objective, solutions with smaller constraint violation values can also be chosen as non-dominated solutions. In this way, we can optimize objectives and handle constraints simultaneously.

Afterward, a truncation method is needed to maintain the original population size N . To this end, solutions in the first k levels are chosen, where k represents the maximum value satisfying $|F_1 \cup F_2 \cup \dots \cup F_k| < N$. Then, $|F_1 \cup F_2 \cup \dots \cup F_k \cup F_{k+1}| - N$ solutions need to be selected at the $k + 1$ level based on association between solutions and the predefined reference points (see Algorithm 4). This process is the same as Deb's work [10]. In line 4 of Algorithm 4, the normalization of objectives only performs from ϵ -feasible solutions. It is worth noting that in the normalization of objectives and the association operation, we no longer consider the constraint violation objective. This is because infeasible solutions with smaller constraint violation values have been picked up in the non-dominated sorting process, the goal of the reference-point-based selection is to choose solutions that are near the reference line of each reference point to keep the good distribution of solutions. If we still consider the constraint violation in this procedure, solutions are likely to be unevenly distributed in the objective space. It is also worth to note that the reference-point-based elite selection procedure only considers the original m objectives. So, no additional reference points need to be set for the constraint violation.

D. Computational Complexity

It is evident that the proposed DCNSGA-III does not cost significant computational resources. The additional computational time complexity is non-dominated sorting in $O((m + 1)N^2)$, where m is the number of objectives. Considering that the computational time complexity of the original NSGA-III is $O(mN^2)$, the additional computational time complexity caused by the proposed method is acceptable.

V. EXPERIMENTAL RESULTS

A. Experimental settings

1) *Test suites*: In this section, DCNSGA-III was evaluated on five sets of CMOPs and CMAOPs with no less than three objectives: CF suite [11], MW suite [13], DAS-CMOP suite [14], C-DTLZ suite [12], and DC-DTLZ suite [8]. These problems are very challenging to solve.

The Pareto set of CF suite is highly nonlinear. An algorithm is hard to converge to the true constrained PF when solving the CF suite. The C-DTLZ suite is based on the DTLZ test suite [40] by adding four kinds of constraints in the objective space. The objective number of the C-DTLZ suite is scalable. On the basis of the C-DTLZ suite, the DC-DTLZ suite introduces three types of constraints in the decision space. The DC-DTLZ suite involves many local optima in infeasible regions, an

Algorithm 4: Reference-point-based non-dominated sorting to select N solutions

Input: ϵ -feasible set \mathcal{S}_1 , population size N , reference points Z

Output: \mathcal{P}_{t+1}

- 1 Performing non-dominated-sorting on \mathcal{S}_1 based on $(f_1(\mathbf{x}), \dots, f_m(\mathbf{x}), cv(\mathbf{x}))$ to form the different non-domination levels (F_1, F_2, \dots) ;
- 2 $\mathcal{P}_{t+1} = F_1 \cup F_2 \cup \dots \cup F_k$, where k is the maximum number that satisfying $|F_1 \cup F_2 \cup \dots \cup F_k| \leq N$;
- 3 **while** $|\mathcal{P}_{t+1}| < N$ **do**
- 4 Normalize $(f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))$;
- 5 Associate each solution of \mathcal{P}_{t+1} with closest reference point by the perpendicular distance, compute the niche count of each reference point;
- 6 Select randomly a point r which has the smallest niche count;
- 7 Let I_r be a set of solutions associated with r , $I_r \subseteq F_{k+1}$;
- 8 **if** $|I_r| = 0$ **then**
- 9 Remove r from Z temporarily at current state;
- 10 **else**
- 11 **if** nichecount(r) = 0 **then**
- 12 Select the solution \mathbf{x} which has the smallest perpendicular distance to r ;
- 13 **else**
- 14 Select a solution \mathbf{x} randomly from I_r ;
- 15 **end**
- 16 $\mathcal{P}_{t+1} = \mathcal{P}_{t+1} \cup \{\mathbf{x}\}$, nichecount(r) = nichecount(r) + 1, $F_{k+1} = F_{k+1} \setminus \mathbf{x}$.
- 17 **end**
- 18 **end**
- 19 **return** \mathcal{P}_{t+1} .

algorithm may be easy to be trapped on some infeasible local optima and cannot enter feasible areas. The MW suite was proposed very recently which contains diverse characteristics, e.g., small feasibility regions, multiple complex nonlinear constraints, scalable number of objectives, and high-dimensional decision vectors. The DAS-CMOP suite involves three types of hardness: feasibility-hardness, convergence-hardness, and diversity-hardness. The hardness of these three types can be adjustable.

Note that some of the above mentioned test suites include two and three objectives problems (e.g., CFs, MWs, and DAS-CMOPs), we only choose those problems who have no less than three objectives ($m \geq 3$) as benchmarks.

2) *Performance indicators*: To compare the performance of different algorithms, two widely-used indicators were adopted in our experiments:

Inverted Generational Distance (IGD) [41]: it measures the average distance from the points in the true PF P^* to their closest solution in the obtained non-dominated set P :

$$IGD(P^*, P) = \frac{\sum_{\mathbf{x} \in P^*} \text{dis}(\mathbf{x}, P)}{|P^*|} \quad (8)$$

where $\text{dis}(\mathbf{x}, P)$ is the Euclidean distance between \mathbf{x} and its

nearest neighbor in P . A low IGD value indicates that the approximate PF is close to the true PF as well as having a good distribution, which could measure both the convergence and diversity of solutions. In our experiment, approximately 10,000 uniformly distributed points sampled on each constrained Pareto front are used as the reference points for calculating IGD.

Hypervolume (HV) [42]: it measures the volume of the objective space enclosed by the obtained solution set and a reference point \mathbf{z}^r :

$$HV(P) = VOL(\bigcup_{\mathbf{x} \in P} [x_1, z_1^r] \times \cdots \times [x_m, z_m^r]) \quad (9)$$

where VOL represents the Lebesgue measure. A large HV value indicates a good dominance relation. In the calculation of HV, a crucial issue is the choice of the reference point. Choosing a reference point, which is slightly larger than the worst value of each objective on the PF, is suitable since the effects of convergence and diversity of the set can be well balanced. In our experiments, we set the reference point \mathbf{z}^r to $(1.1, \dots, 1.1)^T$ after normalizing the objective values to $[0,1]$ with the range of each objective (i.e., 1.1 times of the upper bounds of the normalized PF [43]). Note that solutions with the normalized values larger than 1.1 in any objective are abandoned, and thereby contributing zero to the calculation of HV.

The above two indicators only consider feasible solutions for the final population. When an algorithm could not find a feasible solution over 30 runs consistently, the mean and standard deviation of IGD and HV values are replaced by the symbol “Infeasible”.

Single-problem Wilcoxon rank-sum test at 0.05 significance level was carried out on each test problem for indicating the significant differences between the two algorithms. The symbols +, −, and \approx means that the proposed DCNSGA-III performs significantly better than, significantly worse than, and statistically equivalent to the competitor, respectively. Additionally, the multi-problem Wilcoxon signed rank test at a 0.05 significance level is conducted to distinguish the differences between a pair of algorithms on all problems. A final ranking of all algorithms is given according to the Friedman test.

3) *Algorithms for comparison*: The following six state-of-the-art CMAOEs were chosen for performance comparison: A-NSGA-III [12], C-MOEA/D [12], I-DBEA [36], PPS [35], C-TAEA [8] and C-AnD [18]. Note that A-NSGA-III, C-MOEA/D, I-DBEA, and C-AnD all adopt CDP to compare pairs of solutions, while I-DBEA, C-TAEA, and PPS utilize the information of infeasible solutions to some extent.

4) *Parameter Settings*: For DCNSGA-III, the generation of reference points are in the same way as [10]. Table I presents the number of reference points and population size that DCNSGA-III used. The control parameter cp in Eq. (6) was set to 5. The simulated binary crossover (SBX) [16] and polynomial mutation (PM) [16] were used as the reproduction operators with parameters set as follows:

- SBX: crossover probability $p_c=0.9$ and distribution index $\eta_c=30$;

TABLE I
NUMBER OF REFERENCE POINTS AND CORRESPONDING POPULATION SIZE.

| m | Reference No. | Popsiz |
|-----|---------------|--------|
| 3 | 91 | 92 |
| 5 | 210 | 212 |
| 8 | 156 | 156 |
| 10 | 275 | 276 |
| 15 | 135 | 136 |

TABLE II
STATISTICS OF PERFORMANCE COMPARISONS OF DCNSGA-III WITH ITS COMPETITORS ACCORDING TO IGD AND HV INDICATORS, RESPECTIVELY.

| IGD | + / − / \approx | R^+ | R^- | $\alpha=0.05$ |
|--------------------------|-------------------|--------|-------|---------------|
| DCNSGA-III vs A-NSGA-III | 44/6/14 | 1788.5 | 227.5 | Yes |
| DCNSGA-III vs C-MOEA/D | 43/14/7 | 1595.0 | 485.0 | Yes |
| DCNSGA-III vs I-DBEA | 47/1/6 | 2014.0 | 2.0 | Yes |
| DCNSGA-III vs PPS | 51/11/2 | 1761.0 | 319.0 | Yes |
| DCNSGA-III vs C-TAEA | 38/13/13 | 1335.0 | 745.0 | Yes |
| DCNSGA-III vs C-AnD | 41/14/9 | 1578.5 | 501.5 | Yes |
| HV | + / − / \approx | R^+ | R^- | $\alpha=0.05$ |
| DCNSGA-III vs A-NSGA-III | 41/4/19 | 1806.5 | 209.5 | Yes |
| DCNSGA-III vs C-MOEA/D | 43/11/10 | 1724.5 | 291.5 | Yes |
| DCNSGA-III vs I-DBEA | 58/0/6 | 2011.0 | 5.0 | Yes |
| DCNSGA-III vs PPS | 47/8/9 | 1683.5 | 332.5 | Yes |
| DCNSGA-III vs C-TAEA | 28/18/18 | 1300.5 | 715.5 | Yes |
| DCNSGA-III vs C-AnD | 35/16/13 | 1581.0 | 499.0 | Yes |

“+”, “−”, and “ \approx ” denote the number of the performance of DCNSGA-III is significantly better than, worse than, and equal to its peers, respectively. R^+ , R^- represent the sum of ranks, $R^+ > R^-$ means that the algorithm of this paper is better than the compared algorithm and vice versa, “Yes” signifies the proposed DCNSGA-III is significantly better than the corresponding algorithm at a 0.05 significance level.

- PM: mutation probability $p_m=1/n$ and distribution index $\eta_m=20$.

For fair comparisons, we adopted the recommended parameter settings for the compared algorithms that have achieved the best performance as reported in the literature. The maximum function evaluations are presented in Table S-II of the supplementary document. All the compared algorithms were implemented in PlatEMO [44].

B. Comparison with six state-of-the-art CMAOEs

Due to the space limitation, the detailed average and standard deviation results over 30 runs in terms of IGD indicator and HV indicator are presented in Table S-III and Table S-IV of the supplementary file. The best results for each instance are highlighted with a gray background. The scatter plots of the final obtained PF with the median IGD values over 30 runs obtained by all algorithms were plotted in the supplementary file, where an approximated PF is shown in blue and the true constrained PF is shown in purple.

The CF test instances have nonlinear Pareto optimal solutions in the decision space, it is hard for an algorithm to converge to its true constrained PF. Furthermore, the feasible ratio of CF8 and CF10 test instances are very small. Only C-TAEA, PPS, and DCNSGA-III can find feasible solutions consistently for the CF test instances, while others struggle in searching for feasible regions.

For DC1-DTLZ1 and DC1-DTLZ3 problems, the constraints split the feasible region into three disjoint tapered

strips. The obtained non-dominated solutions in Figs. S-16-S-17 of the supplementary file indicate that A-NSGA-III, C-MOEA/D, and I-DBEA missed some feasible PF segments, since the population has the risk of being trapped in a local feasible region and all feasible regions cannot be found easily. Overall, C-TAEA and C-AnD perform best on the DC1-DTLZ1 problem and DC1-DTLZ3 problem, respectively.

For C1-DTLZ1, DC2-DTLZ1, and DC2-DTLZ3 problems, only a small portion of an area close to the PF is feasible. We can see that DCNSGA-III and C-TAEA have the best overall performance on this type of problem, while others hardly find feasible solutions on most instances, i.e., A-NSGA-III, C-MOEA/D, I-DBEA, PPS, and C-AnD were stuck at some infeasible areas on both DC2-DTLZ1 and DC2-DTLZ3 test instances. This is because the characteristics of these problems are easy to make the population stagnate in locally infeasible optimal, an algorithm is hard to escape from the trapped areas which are surrounded by infeasible blocks.

For DC3-DTLZ1 and DC3-DTLZ3 problems, the feasible area is split into several sections, and there is a sheet feasible area above the PF. The results suggest that C-TAEA and the proposed DCNSGA-III have an advantage in solving this kind of problem. The superiority of DCNSGA-III can be attributed to the cooperation between handling constraints and optimizing objectives. The PF of this kind of problems consists of many separate parts, at each generation, DCNSGA-III focuses on searching for its PF of the current generation, which can maintain the completeness of its PF till the end of the run.

Notice that in this paper, the reference points are predefined and distributed uniformly in a unit simplex. DCNSGA-III with this setting of reference points may not work well on problems with irregular PF. This can be reflected by figures of final obtained approximated solutions in the supplementary document, e.g., MW14 problem with scaled objectives, C2-DTLZ2 problem with disconnected PF, C3-DTLZ4 with a mixed nature of PF. DCNSGA-III cannot obtain evenly-distributed non-dominated solutions on these test problems.

The statistical results on the basis of the IGD and HV indicators are summarized in Table II. In the light of the IGD indicator, we can see from the results that DCNSGA-III is significantly better than its competitors on at least 38/64 and at most 51/64 test problems. The Multi-problem Wilcoxon signed-rank test statistical results indicate that DCNSGA-III provides higher R^+ values than R^- values in all cases, and also performs significantly better than its competitors at a 0.05 significance level in all cases. Fig. 4 reveals the statistical rank obtained by the Friedman test of all algorithms for all problems, where DCNSGA-III ranks first among seven algorithms in terms of the IGD indicator.

According to the HV indicator, as described in Table II, DCNSGA-III has an edge over A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, and C-AnD on 41, 43, 58, 47, 28, and 35 test functions, respectively, which is larger than the number of test instances where the performance of DCNSGA-III is worse than its peer algorithms. The multiple-problem Wilcoxon's signed-rank test suggest that DCNSGA-III provides higher R^+ values than R^- values in all cases, and

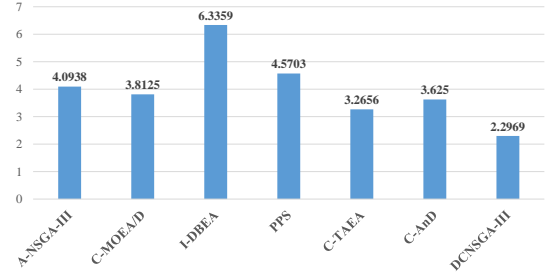


Fig. 4. Friedman's test among seven CMAOEs in terms of IGD. The smaller the ranking, the better the performance of an algorithm.

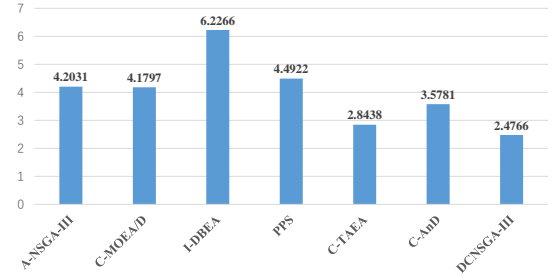


Fig. 5. Friedman's test among seven CMAOEs in terms of HV. The smaller the ranking, the better the performance of an algorithm.

performs significantly better than its six competitors. Furthermore, DCNSGA-III achieves the first rank regarding the Friedman's test in Fig. 5.

Overall, from the above comparison, we can conclude that the proposed DCNSGA-III exhibits better performance than the six competitors on solving CMAOPs tested in this paper. Moreover, the superiority of DCNSGA-III against the six competitors particularly is shown on problems with a small ratio of feasible solutions and complex constraints.

VI. FURTHER INVESTIGATION

A. Investigation of the ϵ feasibility of the population

In Section IV, the union of the parent population and the offspring population is divided into two groups based on their ϵ feasibility: the ϵ -feasible set S_1 and the ϵ -infeasible set S_2 . To investigate the ϵ feasibility of the merged population, two test problems CF10 and DC3-DTLZ1, were tested for DCNSGA-III. Fig. 6 shows the changes in their ϵ -feasible ratio over time.

From Fig. 6, we can have the following observations:

- In terms of the ϵ feasibility, the number of ϵ -feasible solutions is consistently larger than the population size, which means the population can search at the relatively large ϵ -feasible search space, so the population can focus on the tradeoffs between diversity and convergence in each generation. In addition, such a large number of ϵ -feasible solutions in each generation also verify the effectiveness of the designed tournament selection operator, which always gives priority to the selection of ϵ -feasible solutions.

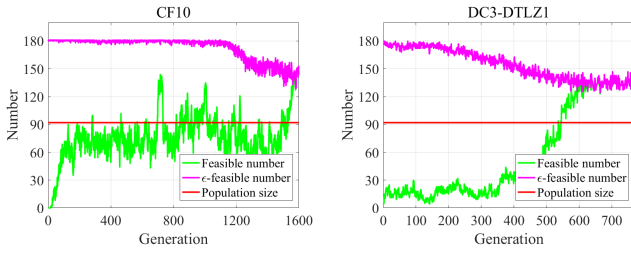


Fig. 6. ε feasibility of the population obtained by DCNSGA-III on CF10 problem and DC3-DTLZ1 problem, respectively.

- In terms of feasibility, the number of feasible solutions fluctuates over time, which means the population shuttles between feasible and infeasible areas. The above phenomena suggest that DCNSGA-III is not only driven by feasibility. Instead, it is driven by both objectives and constraints, which helps it cross the local feasible and infeasible areas.
- At the end of the evolutionary stage, the number of ε -feasible solutions is approximately equal to the number of feasible solutions, which indicates that the ε constraint boundary shrinks to the original constrained boundary, the Pareto optimal solutions of the transformed problem are the Pareto optimal solutions of the original problem, thus DCNSGA-III can find the Pareto optimal solutions of the original CMaOP without the affect of constraints during the search.

B. Investigation of the search behavior

For the ease of visualizing the working mechanism of DCNSGA-III, we select three constrained bi-objective optimization problems (i.e., MW1, MW3, and MW9 [13]) to visualize the movement of solutions obtained by DCNSGA-III. For MW1, the constraints make its PF partially feasible. For MW3, constraints make a part of the unconstrained PF infeasible and a part of solutions on the boundary of the feasible region become the Pareto optimal solutions. For MW9, the unconstrained PF is entirely located outside the feasible region. Thus, the constrained PF is composed of a part of the boundary of the feasible region. The maximum number of generations and the population size were set to 600 and 100, respectively. Other parameter settings were kept unchanged. It is worth noting that only non-dominated solutions are plotted.

For the MW1 problem, as shown in Fig. 7, the population first approaches the unconstrained PF with good distribution. As the dynamic ε constrained boundary gradually shrunk to the original constraint boundary, infeasible solutions are driven to move to their nearest feasible segments.

Similarly, for the MW3 problem, as shown in Fig. 8, at first, the population gets to its approximated unconstrained PF. In the middle and late stages of the evolution, some feasible solutions keep unchanged due to some parts of the unconstrained PF are segments of the constrained PF. Other infeasible solutions move back to the segments of the constrained PF.

For the MW9 problem, as shown in Fig. 9, the population firstly crosses the large infeasible areas and reaches the uncon-

strained PF. Since the constrained PF is on the boundary of the feasible region, the population returns to the constrained PF from infeasible regions.

Combined with the changes of ε dynamic constraint boundary as the number of generations (environment state) increases in Fig. 2, we can make the following conclusions for the above phenomenon:

- 1) In the early stage of the evolution, the ε dynamic constraint boundary shrinks very slowly. In this case, DCNSGA-III searches Pareto-optimal solutions of $m+1$ objectives in all space, which is beneficial to help the search to cross over the large, multi-modal, and discrete infeasible regions, and forces the population to towards the unconstrained PF. This is also helpful to deal with the situation where the constrained PF is the same as or part of the unconstrained PF.
- 2) In the middle stage of the evolution, the ε dynamic constraint boundary drops very fast. Under the increasing pressure of the ε constraint boundary, the search switches from optimality priority to feasibility priority.
- 3) In the later stage of the evolution, the ε dynamic constraint boundary is close to 0, which means the relaxed constraint boundary shrinks to its original constraint boundary. DCNSGA-III searches Pareto-optimal solutions of m objectives mainly in the feasible space.

C. Comparison with five representative constraint-handling techniques under NSGA-III

As we all know, CMAOEs consist of two basic components: MaOEs and constraint-handling techniques. These two parts all have a great influence on the performance of CMAOEs for solving CMaOPs. MaOEs have different performance on distinct test problems [45]. Therefore, to investigate the effectiveness of DCNSGA-III on constraint handling, an additional experiment was conducted, where five well-known constraint-handling techniques: CDP [16], SR [24], self-adaptive penalty (SP) [46], ε method [47], and adaptive trade-off model (ATM) [48] were embedded into the same framework of NSGA-III, namely CDP-NSGA-III, SR-NSGA-III, SP-NSGA-III, ε -NSGA-III, and ATM-NSGA-III, respectively. The parameter settings of these five algorithms are listed in Section S-VI of the supplementary file. The DC2-DTLZ1 and DC2-DTLZ3 instances were selected as benchmarks because they exhibit complex constraint features, such as small feasible regions, and multi-modal of the constraint violation. So, they could provide a comprehensive assessment of the performance of different constraint-handling techniques.

Table III presents the average and standard deviation of IGD values of the approximation sets obtained by each algorithm over 30 independent runs. From Table III, we can see that CDP-NSGA-III, SP-NSGA-III, ε -NSGA-III, and ATM-NSGA-III cannot find feasible solutions in all runs. SR-NSGA-III can achieve feasible runs on one case, this can be attributed to the fact that the probability parameter P_f in SR was set to 0.45, which means there is a 45% probability of comparing two solutions based on objective values. This can avoid the search to fall into the constraint-violation's

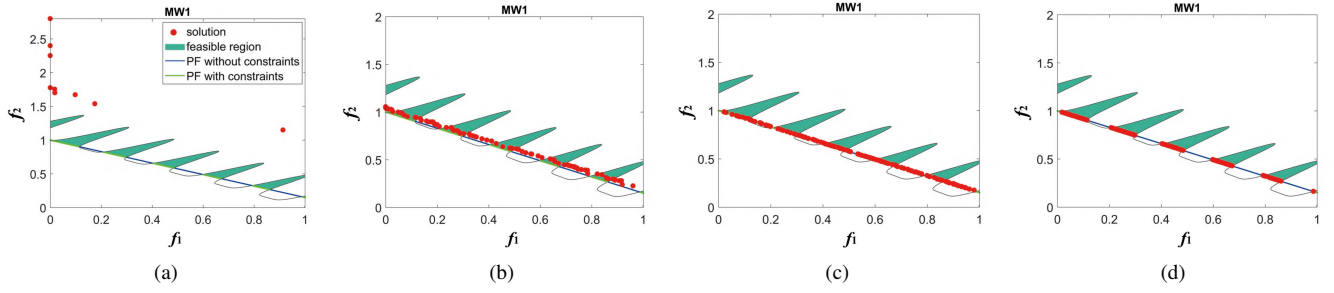


Fig. 7. Visualization of solutions' movement on MW1 problem. (a) generation=50; (b) generation=200; (c) generation=300; (d) generation=550.

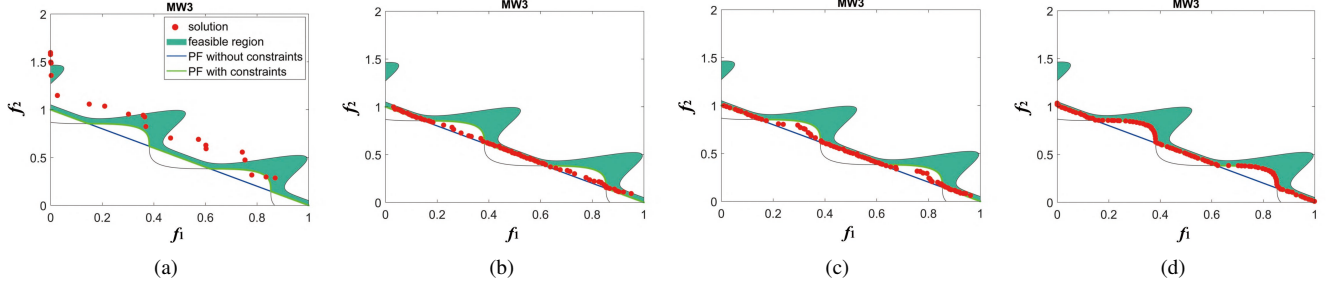


Fig. 8. Visualization of solutions' movement on MW3 problem. (a) generation=50; (b) generation=150; (c) generation=300; (d) generation=550.

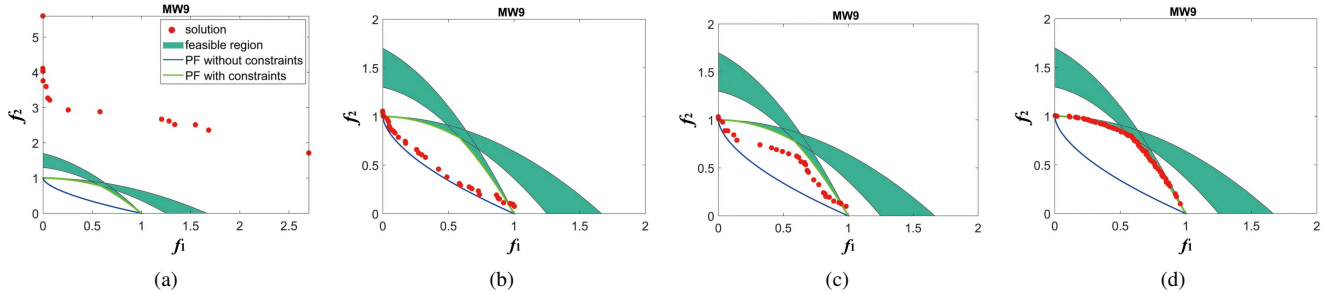


Fig. 9. Visualization of solutions' movement on MW9 problem. (a) generation=50; (b) generation=300; (c) generation=400; (d) generation=550.

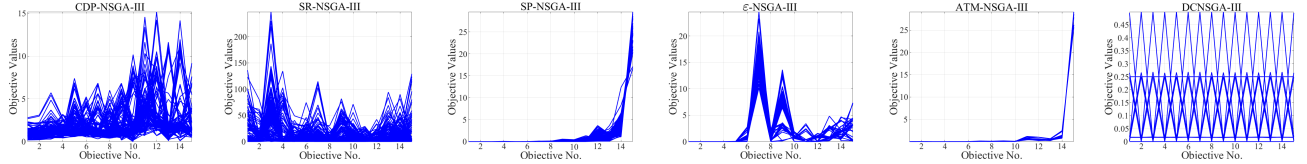


Fig. 10. Parallel coordinates of obtained non-dominated solutions for DC2-DTLZ1 problem with 15 objectives using CDP-NSGA-III, SR-NSGA-III, SP-NSGA-III, ϵ -NSGA-III, ATM-NSGA-III, and DCNSGA-III from left to right (in median IGD value).

local optima to some extent. Although ϵ -NSGA-III and the proposed DCNSGA-III all relax the original feasible region, ϵ -NSGA-III is not able to provide enough selection pressure to pull the population to the boundary of the feasible region. For instance, if the ϵ -feasible region contains some candidate solutions having better objective values than those of the global optima, the population may converge to these solutions, and thus they are trapped into local optima in the infeasible region. DCNSGA-III considers the constraint violation as an extra objective, which can push the population towards the regions with smaller constraint violations. That is why it can achieve feasible solutions on nine out of ten test cases in Table III. This is also demonstrated from the parallel coordinates in Fig. 10 on the 15-objective DC2-DTLZ1 instance, where

all other peer algorithms are trapped in the infeasible region far away from the constrained PF, while the non-dominated solutions obtained by DCNSGA-III can evenly distribute on the constrained PF.

To sum up, the above comparison demonstrates that the proposed problem transformation method exhibits better performance than the five representative constraint-handling techniques in solving DC2-DTLZ test instances.

VII. CONCLUSION AND FUTURE DIRECTIONS

The major challenge of solving CMaOPs is dealing with the interrelationship between objectives and constraints in the high-dimensional objective space. To optimize conflicting objectives and tackle constraints simultaneously, this paper proposed a problem transformation technique which transforms

TABLE III

COMPARATIVE RESULTS OF IGD VALUES (MEAN AND STANDARD DEVIATION) OBTAINED BY CDP-NSGA-III, SR-NSGA-III, SP-NSGA-III, ϵ -NSGA-III, ATM-NSGA-III, AND DCNSGA-III ON DC2-DTLZ1 AND DC2-DTLZ3 INSTANCES.

| Problem | m | CDP-NSGA-III | SR-NSGA-III | SP-NSGA-III | ϵ -NSGA-III | ATM-NSGA-III | DCNSGA-III |
|-----------|-----|--------------|---------------------|-------------|----------------------|--------------|--------------------|
| DC2-DTLZ1 | 3 | Infeasible | 3.6014e-1(3.93e-2)+ | Infeasible | Infeasible | Infeasible | 2.0562e-2(4.76e-5) |
| | 5 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 5.2710e-2(8.86e-5) |
| | 8 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 9.8276e-2(5.58e-3) |
| | 10 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 1.2312e-1(2.21e-3) |
| | 15 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 1.9088e-1(1.29e-2) |
| DC2-DTLZ3 | 3 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 5.4566e-2(1.29e-4) |
| | 5 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 1.6515e-1(1.95e-5) |
| | 8 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 3.8180e-1(8.92e-2) |
| | 10 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 4.7193e-1(6.85e-2) |
| | 15 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible |

“+” represents the performance of DCNSGA-III is significantly better than the other peers according to the Wilcoxon’s rank sum test at a 0.05 significance level; “Infeasible” stands for the algorithm cannot find feasible solutions in all 30 runs.

an m -objective CMAOP into an equivalent $(m+1)$ -objective DCMAOP. After problem transformation, the difficulty of solving a highly COP in the many-objective space is converted to solve a DCMAOP. The transformation is helpful for an algorithm to get across the broad, separated, and multi-modal infeasible regions. Then, a MaOEA can be applied to search for good solutions with tradeoff among convergence, diversity, and constraint violation.

A specialized non-dominated sorting algorithm (DCNSGA-III) based on the problem transformation model was proposed. The key idea is to keep most solutions of the population ϵ feasible. To this end, in DCNSGA-III, a mating selection mechanism was designed to select the potential mating parents, a modified environment selection operator combined with the ϵ -constrained dominance principle was developed to classify the population into different non-dominated fronts and choose solutions with better convergence, diversity and constraint violation. To demonstrate the strong competitiveness of the proposed DCNSGA-III, we performed an extensive experimental comparison of DCNSGA-III with six state-of-the-art CMAOEs. A series of benchmark problems for 3, 5, 8, 10, and 15 objectives were chosen to challenge the different abilities of the algorithms. The comparison results reveal that the proposed DCNSGA-III works well on majority CMAOPs, particularly on problems with complex constraints and small feasible regions. Furthermore, the effectiveness of the proposed problem transformation technique has also been validated by comparing it with five well-known constraint-handling techniques.

Due to the existence of constraints, the constrained PF of a test problem may become irregular, i.e., disjoint, degenerate, scaled. The predefined reference vectors will deteriorate the performance for such problems with irregular PF [49], [50]. Therefore, designing adaptive reference points for problems with irregular PF will be our future work.

REFERENCES

- [1] X. Cai, H. Sun, Q. Zhang, and Y. Huang, “A grid weighted sum pareto local search for combinatorial multi and many-objective optimization,” *IEEE Transactions on Cybernetics*, vol. 49, no. 9, pp. 3586–3598, 2019.
- [2] M. Elarbi, S. Bechikh, A. Gupta, L. B. Said, and Y.-S. Ong, “A new decomposition-based nsga-ii for many-objective optimization,” *IEEE Transactions on Systems, Man, and Cybernetics Systems*, vol. 48, no. 7, pp. 1191–1210, 2017.
- [3] J. Wang, T. Weng, and Q. Zhang, “A two-stage multiobjective evolutionary algorithm for multiobjective multidepot vehicle routing problem with time windows,” *IEEE Transactions on Cybernetics*, vol. 49, no. 7, pp. 2467–2478, 2018.
- [4] Y. Zhang, Y. Gong, T. Gu, H. Yuan, W. Zhang, S. Kwong, and J. Zhang, “Decal: Decomposition-based coevolutionary algorithm for many-objective optimization,” *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 27–41, 2019.
- [5] Q. Xu, S. Zeng, F. Zhao, R. Jiao, and C. Li, “On formulating and designing antenna arrays by evolutionary algorithms,” *IEEE Transactions on Antennas and Propagation*, pp. 1–12, 2020, doi:10.1109/TAP.2020.3016181.
- [6] B. Li, J. Li, K. Tang, and X. Yao, “Many-objective evolutionary algorithms: A survey,” *ACM Computing Surveys*, vol. 48, no. 1–35, p. 13, 2015.
- [7] K. Deb, “Constrained multi-objective evolutionary algorithm,” in *Evolutionary and Swarm Intelligence Algorithms*. Springer, 2019, pp. 85–118.
- [8] K. Li, R. Chen, G. Fu, and X. Yao, “Two-archive evolutionary algorithm for constrained multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2018.
- [9] J. Wang, G. Liang, and J. Zhang, “Cooperative differential evolution framework for constrained multiobjective optimization,” *IEEE Transactions on Cybernetics*, vol. 49, no. 6, pp. 2060–2072, 2018.
- [10] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.
- [11] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, “Multiobjective optimization test instances for the cec 2009 special session and competition,” *University of Essex, and Nanyang technological University, technical report*, vol. 264, 2008.
- [12] H. Jain and K. Deb, “An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.
- [13] Z. Ma and Y. Wang, “Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 6, pp. 972–986, 2019.
- [14] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, “Difficulty adjustable and scalable constrained multi-objective test problem toolkit,” *Evolutionary Computation*, pp. 1–28, 2019.
- [15] E. Mezura-Montes and C. A. C. Coello, “Constraint-handling in nature-inspired numerical optimization: past, present and future,” *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, 2011.
- [16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [17] X. Cai, Z. Yang, Z. Fan, and Q. Zhang, “Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization,” *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2824–2837, 2017.
- [18] Z.-Z. Liu, Y. Wang, and P.-Q. Huang, “And: A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation,” *Information Sciences*, vol. 509, pp. 400–419, 2020.

- [19] Z. Fan, Y. Fang, W. Li, X. Cai, C. Wei, and E. D. Goodman, "Moea/d with angle-based constrained dominance principle for constrained multi-objective optimization problems," *Applied Soft Computing*, vol. 74, pp. 621–633, 2019.
- [20] R. Jiao, S. Zeng, and C. Li, "A feasible-ratio control technique for constrained optimization," *Information Sciences*, vol. 502, pp. 201–217, 2019.
- [21] E. Mezuramontes and C. A. C. Coello, "Useful infeasible solutions in engineering optimization with evolutionary algorithms," in *Mexican International Conference on Artificial Intelligence*, 2005, pp. 652–662.
- [22] L. While and P. Hingston, "Usefulness of infeasible solutions in evolutionary search: An empirical and mathematical study," in *IEEE Congress on Evolutionary Computation*, 2013, pp. 1363–1370.
- [23] H. Singh, K. Alam, and T. Ray, "Use of infeasible solutions during constrained evolutionary search: A short survey," in *Australasian Conference on Artificial Life and Computational Intelligence*, 2016, pp. 193–205.
- [24] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.
- [25] T. Takahama and S. Sakai, "Constrained optimization by the ε -constrained differential evolution with an archive and gradient-based mutation," in *IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–9.
- [26] P. D. Surry, N. J. Radcliffe, and I. D. Boyd, "A multi-objective approach to constrained optimisation of gas supply networks: The comoga method," in *AISB Workshop on Evolutionary Computing*. Springer, 1995, pp. 166–180.
- [27] S. Zeng, R. Jiao, C. Li, and R. Wang, "Constrained optimisation by solving equivalent dynamic loosely-constrained multiobjective optimisation problem," *International Journal of Bio-Inspired Computation*, vol. 13, no. 2, pp. 86–101, 2019.
- [28] S. Zeng, R. Jiao, C. Li, X. Li, and J. S. Alkasassbeh, "A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization," *IEEE Transactions on Cybernetics*, vol. 47, no. 9, pp. 2678–2688, 2017.
- [29] C. Liu, Q. Zhao, B. Yan, S. Elsayed, T. Ray, and R. Sarker, "Adaptive sorting-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 247–257, 2018.
- [30] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2015.
- [31] Y. Zhou, M. Zhu, J. Wang, Z. Zhang, Y. Xiang, and J. Zhang, "Tri-goal evolution framework for constrained many-objective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2018, doi:10.1109/TSMC.2018.2858843.
- [32] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2014.
- [33] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.
- [34] Q. Zhu, Q. Zhang, and Q. Lin, "A constrained multiobjective evolutionary algorithm with detect-and-escape strategy," *IEEE Transactions on Evolutionary Computation*, pp. 1–10, 2020, doi:10.1109/TEVC.2020.2981949.
- [35] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 44, pp. 665–679, 2019.
- [36] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 445–460, 2014.
- [37] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 264–283, 2015.
- [38] Q. Zhu, Q. Lin, J. Li, C. A. C. Coello, Z. Ming, J. Chen, and J. Zhang, "An elite gene guided reproduction operator for many-objective optimization," *IEEE Transactions on Cybernetics*, pp. 1–14, 2019.
- [39] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [40] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *IEEE Congress on Evolutionary Computation*, vol. 1. IEEE, 2002, pp. 825–830.
- [41] P. A. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.
- [42] E. Zitzler, "Evolutionary algorithms for multiobjective optimization: Methods and applications," vol. 63. Citeseer, 1999.
- [43] H. Ishibuchi, R. Imada, Y. Setoguchi, and Y. Nojima, "How to specify a reference point in hypervolume calculation for fair performance comparison," *Evolutionary Computation*, vol. 26, no. 3, pp. 411–440, 2018.
- [44] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.
- [45] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2016.
- [46] W. Y. Gebre, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 514–525, 2009.
- [47] T. Takahama and S. Sakai, "Constrained optimization by the ε -constrained differential evolution with gradient-based mutation and feasible elites," in *IEEE International Conference on Evolutionary Computation*. IEEE, 2006, pp. 1–8.
- [48] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 80–92, 2008.
- [49] M. Asafuddoula, H. K. Singh, and T. Ray, "An enhanced decomposition-based evolutionary algorithm with adaptive reference vectors," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2321–2334, 2018.
- [50] X. Cai, Z. Mei, and Z. Fan, "A decomposition-based many-objective evolutionary algorithm with two types of adjustments for direction vectors," *IEEE Transactions on Cybernetics*, vol. 48, no. 8, pp. 2335–2348, 2018.



Ruwang Jiao is now a PhD candidate in School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan, China. In 2019-2020, he was a visiting scholar in Data Science and Artificial Intelligence Research Center, Nanyang Technological University, Singapore.

His research interests include evolutionary constrained optimization, multi-objective optimization, and their applications in antenna design.



Sanyou Zeng received the M.Sc. degree in mathematics from Hunan University, Changsha, China, in 1995, and the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 2002.

He has been a Professor with the China University of Geosciences, Wuhan, since 2004. His current research interests include evolutionary computation with machine learning for solving problems with constraints, multi-objective, dynamic environments, and expensive costs, especially the antenna design problems.



Changhe Li (M'12) received the B.Sc. and M.Sc. degrees in computer science from China University of Geosciences, Wuhan, China, in 2005 and 2008, respectively, and the Ph.D. degree in computer science from the University of Leicester, U.K. in July 2011.

He was an associate professor with the School of Computer Science, China University of Geosciences, Wuhan, China from 2011 to 2017. He is currently a professor with the School of Automation, China University of Geosciences, Wuhan, China. He is

the chair of the Task Force on Evolutionary Computation in Dynamic and Uncertain Environments. His research interests include intelligent optimization and machine learning.



Shengxiang Yang (M'00–SM'14) received the Ph.D. degree from Northeastern University, Shenyang, China in 1999.

He is currently a Professor in Computational Intelligence and Director of the Centre for Computational Intelligence, School of Computer Science and Informatics, De Montfort University, Leicester, U.K. He has over 300 publications with an H-index of 55 according to Google Scholar. His current research interests include evolutionary computation, swarm intelligence, artificial neural

networks, data mining and data stream mining, and relevant real-world applications. He serves as an Associate Editor/Editorial Board Member of a number of international journals, such as the *IEEE Transactions on Cybernetics*, *IEEE Transactions on Evolutionary Computation*, *Information Sciences*, and *Enterprise Information Systems*.



Yew-Soon Ong (M'99–SM'12–F'18) received the Ph.D. degree in artificial intelligence in complex design from the University of Southampton, U.K., in 2003.

He is President's Chair Professor in Computer Science at Nanyang Technological University (NTU), and serves as Chief Artificial Intelligence Scientist at the Agency for Science, Technology and Research Singapore. At NTU, he serves as Director of the Singtel-NTU Cognitive & Artificial Intelligence Joint Lab, and Director of the Data Science

and Artificial Intelligence Research Center. His research interest in artificial and computational intelligence. He is founding Editor-in-Chief of the *IEEE Transactions on Emerging Topics in Computational Intelligence* and associate editor of *IEEE TNNLS*, the *IEEE Cybernetics*, *IEEE TEVC*, *IEEE TAI* and others. He has received several IEEE outstanding paper awards and was listed as a Thomson Reuters highly cited researcher and among the World's Most Influential Scientific Minds.

Supplementary File for “Handling Constrained Many-objective Optimization Problems via Problem Transformation”

Ruwang Jiao, Sanyou Zeng, Changhe Li, *Member, IEEE*,
Shengxiang Yang, *Senior Member, IEEE*, and Yew-Soon Ong, *Fellow, IEEE*

S-I. PARAMETER SETTINGS

The characteristics of the constrained PF and the maximum number of function evaluations for CF suite [1], MW suite [2], DAS-CMOP suite [3], C-DTLZ suite [4], and DC-DTLZ suite [5] are given in Table S-I and Table S-II, respectively, where m and n are the number of objectives and decision variables, respectively. FES represents the number of function evaluations. In our experiment, some constraint function parameters of these test problems were set as follows:

- For C1-DTLZ3: $r = \{9, 12.5, 12.5, 15, 15\}$ for $m = \{3, 5, 8, 10, 15\}$;
- For C2-DTLZ2: $r = \{0.4, 0.5, 0.5, 0.5, 0.5\}$ for $m = \{3, 5, 8, 10, 15\}$;
- For DC1-DTLZ1 and DC1-DTLZ3 problems: $a=5$, $b = 0.95$;
- For DC2-DTLZ1 and DC2-DTLZ3 problems: $a=3$, $b = 0.9$;
- For DC3-DTLZ1 and DC3-DTLZ3 problems: $a=5$, $b = 0.5$.

S-II. DETAILED RESULTS IN TERMS OF IGD AND HV INDICATORS

Table S-III and Table S-IV list the average and standard deviation of IGD and HV values over 30 runs for seven CMOEAs: A-NSGA-III [4], C-MOEA/D [4], I-DBEA [6], PPS [7], C-TAEA [5], C-AnD [8], and the proposed DCNSGA-III, respectively. The best results for each instance are highlighted with a gray background.

S-III. EFFECTIVENESS OF THE MATING SELECTION

In the high-dimensional objective space, the choice of parents from the entire population is not a good idea since it may degrade the performance of a MOEA [9], [10]. To improve the effects of the genetic operation, in the mating selection operator of DCNSGA-III, an ε -feasible solution was selected over an ε -infeasible solution or a solution with a smaller constraint violation between two ε -infeasible solutions is chosen. In order to verify this mating selection operator is necessary for

TABLE S-I
CHARACTERISTICS OF CONSTRAINED PF OF DIFFERENT TEST PROBLEMS.

| Problem | Geometry | Connectivity |
|-----------|----------|--------------|
| CF8 | Concave | Disconnected |
| CF9 | Concave | Disconnected |
| CF10 | Concave | Disconnected |
| MW4 | Linear | Connected |
| MW8 | Concave | Disconnected |
| MW14 | Scaled | Disconnected |
| DAS-CMOP7 | Linear | Disconnected |
| DAS-CMOP8 | Concave | Disconnected |
| DAS-CMOP9 | Concave | Disconnected |
| C1-DTLZ1 | Linear | Connected |
| C1-DTLZ3 | Concave | Connected |
| C2-DTLZ2 | Concave | Disconnected |
| C3-DTLZ1 | Convex | Connected |
| C3-DTLZ4 | Mixed | Connected |
| DC1-DTLZ1 | Linear | Disconnected |
| DC1-DTLZ3 | Concave | Disconnected |
| DC2-DTLZ1 | Linear | Connected |
| DC2-DTLZ3 | Concave | Connected |
| DC3-DTLZ1 | Linear | Disconnected |
| DC3-DTLZ3 | Concave | Disconnected |

DCNSGA-III, an additional experiment was executed. We compare two other mating selection operators. The first one is the same as C-NSGA-III and A-NSGA-III [4], where a feasible solution will be chosen over an infeasible one, or a solution with a smaller constraint violation will be selected between two infeasible solutions, or pick one at random from two feasible solutions. We denote this operator as *Variant2*. The other mating selection is randomly choosing one from two solutions in binary tournament selection regardless of their constraint violation, and this operator is referred as *Variant3*. The mating selection operator in the proposed DCNSGA-III is called *Variant1*. We compare the feasibility ratio of these three operators. The feasibility ratio equals the

TABLE S-III
COMPARATIVE RESULTS OF IGD VALUES (MEAN AND STANDARD DEVIATION) OBTAINED BY A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AND, AND DCNSGA-III, RESPECTIVELY.

| Problem | m | A-NSGA-III | C-MOEA/D | I-DBEA | PPS | C-TAEA | C-AND | DCNSGA-III |
|-----------|-----|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|---------------------|
| CF8 | 3 | Infeasible | Infeasible | Infeasible | 1.1713e-1 (1.06e-2) — | 3.8116e-1 (1.33e-1) + | Infeasible | 2.6301e-1 (1.02e-1) |
| CF9 | 3 | 1.7495e-1 (1.18e-1) ≈ | 9.0134e-2 (6.69e-3) — | 5.6681e-1 (2.03e-1) + | 5.1617e-2 (6.90e-3) — | 1.1855e-1 (1.95e-2) ≈ | 1.1281e-1 (2.64e-2) ≈ | 1.6353e-1 (8.86e-2) |
| CF10 | 3 | Infeasible | Infeasible | Infeasible | 1.8890e-1 (7.75e-2) — | 5.2363e-1 (1.10e-1) + | Infeasible | 3.0136e-1 (9.59e-2) |
| MW4 | 3 | 4.3700e-2 (1.44e-3) + | 6.0256e-2 (5.41e-3) + | 5.9749e-1 (1.78e-1) + | Infeasible | 4.2191e-2 (1.49e-4) ≈ | 4.9214e-2 (1.13e-2) + | 4.2514e-2 (8.49e-4) |
| MW8 | 3 | 8.1382e-2 (1.09e-1) ≈ | 5.1357e-2 (2.61e-3) + | 7.9668e-1 (1.58e-1) + | 1.3604e-1 (6.24e-2) + | 4.9345e-2 (2.58e-3) ≈ | 1.8619e-1 (1.38e-1) + | 4.9645e-2 (2.15e-3) |
| MW14 | 3 | 1.2103e-1 (2.09e-2) — | 2.1889e-1 (3.00e-2) + | 2.7402e+0 (6.39e-1) + | 2.4732e-1 (8.36e-2) + | 1.0784e-1 (7.81e-3) — | 4.3795e-1 (1.66e-1) + | 1.3346e-1 (3.64e-3) |
| DASCMP7 | 3 | 5.1396e-2 (8.71e-3) ≈ | 1.2545e-1 (1.39e-1) + | 1.1370e+0 (2.94e-1) + | 9.2612e-2 (9.25e-2) + | 4.3713e-2 (2.02e-2) — | 4.5122e-2 (2.18e-2) — | 4.8872e-2 (9.72e-3) |
| DASCMP8 | 3 | 1.2892e-1 (1.59e-1) + | 1.4310e-1 (1.06e-1) + | 1.1315e+0 (2.31e-1) + | 1.1646e-1 (5.06e-2) + | 7.3195e-2 (2.88e-2) + | 5.1216e-2 (3.36e-3) ≈ | 5.1206e-2 (3.50e-3) |
| DASCMP9 | 3 | 3.3171e-1 (3.60e-2) ≈ | 9.9844e-2 (9.13e-3) — | 6.4437e-1 (7.38e-2) + | 4.6456e-2 (5.87e-3) — | 1.6879e-1 (3.21e-2) — | 3.0212e-1 (9.26e-2) — | 3.9821e-1 (1.59e-1) |
| C1-DTLZ1 | 3 | 2.2535e-2 (1.68e-3) + | 2.0813e-2 (3.40e-4) + | 4.1450e-1 (7.19e-2) + | 3.0376e-2 (6.72e-3) + | 2.3287e-2 (6.84e-4) + | 2.2823e-2 (3.74e-4) + | 2.0375e-2 (3.21e-4) |
| | 5 | 5.2074e-2 (4.44e-4) ≈ | 5.2288e-2 (1.58e-4) + | 4.5171e-1 (8.68e-2) + | Infeasible | 5.9592e-2 (4.45e-4) + | 5.3650e-2 (3.70e-4) + | 5.2082e-2 (3.98e-4) |
| | 8 | 1.0243e-1 (1.36e-2) + | 9.3252e-2 (3.29e-4) — | 4.8486e-1 (7.87e-2) + | 1.1148e-1 (1.59e-2) + | 1.2115e-1 (9.64e-4) + | 1.0898e-1 (1.74e-2) + | 9.7726e-2 (9.32e-3) |
| | 10 | 1.1634e-1 (1.74e-2) + | 1.0036e-1 (3.68e-4) — | 4.8149e-1 (9.52e-2) + | 1.2379e-1 (4.08e-3) + | 1.3979e-1 (1.70e-3) + | 1.1292e-1 (9.06e-4) + | 1.0917e-1 (5.09e-3) |
| | 15 | 1.8818e-1 (1.32e-2) + | 1.2703e-1 (7.25e-4) — | 5.4833e-1 (4.11e-2) + | 1.5581e-1 (1.66e-2) — | 2.1419e-1 (5.36e-3) + | 1.6736e-1 (2.79e-3) — | 1.7643e-1 (1.13e-2) |
| C1-DTLZ3 | 3 | 1.0786e+0 (1.39e+0) — | 1.5652e+0 (4.38e+0) + | 2.3378e+0 (1.24e+0) + | 6.6099e-1 (1.66e+0) — | 1.4430e+0 (1.49e+0) + | 2.5902e+0 (1.07e+0) + | 1.2695e+0 (1.45e+0) |
| | 5 | 1.7576e-1 (1.30e-2) + | 5.0539e-1 (1.02e+0) + | 1.6531e-1 (3.58e-4) ≈ | 6.6550e-1 (9.13e-1) + | 3.5807e-1 (5.63e-2) + | 1.7861e-1 (4.84e-3) + | 1.6520e-1 (5.71e-5) |
| | 8 | 5.3340e-1 (1.27e-1) + | 5.7731e-1 (5.74e-1) + | 1.2085e+0 (2.97e-2) + | 5.6555e-1 (3.48e-2) + | 5.1777e-1 (2.08e-2) + | 6.4301e-1 (3.56e-1) + | 3.7026e-1 (1.14e-1) |
| | 10 | 6.2231e-1 (6.86e-2) + | 4.7150e-1 (4.52e-3) + | 1.2417e+0 (2.18e-6) + | 6.6143e-1 (2.01e-2) + | 5.4068e-1 (1.43e-2) + | 5.5324e-1 (5.70e-2) + | 4.7022e-1 (8.33e-2) |
| | 15 | 1.7201e+0 (1.28e+0) + | 6.5922e-1 (6.27e-3) — | 1.2896e+0 (1.23e-5) + | 8.6588e-1 (5.85e-2) + | 7.9537e-1 (2.76e-1) ≈ | 1.5927e+0 (7.63e-1) + | 7.3584e-1 (3.35e-2) |
| C2-DTLZ2 | 3 | 4.4212e-2 (5.23e-4) — | 6.4049e-2 (3.14e-3) + | 8.3458e-1 (1.64e-1) + | 3.3002e-2 (5.84e-4) — | 5.6375e-2 (1.98e-3) + | 6.3248e-2 (8.73e-2) + | 4.8415e-2 (4.50e-4) |
| | 5 | 1.4334e-1 (4.64e-2) + | 1.6285e-1 (2.80e-3) + | 1.0622e+0 (1.88e-1) + | 2.8406e-1 (4.29e-2) + | 1.4676e-1 (1.08e-3) + | 1.3854e-1 (1.05e-3) — | 1.3933e-1 (1.25e-3) |
| | 8 | 3.6654e-1 (2.19e-1) + | 4.4142e-1 (2.55e-1) + | 1.2439e+0 (1.58e-1) + | 4.5042e-1 (7.50e-2) + | 2.3792e-1 (1.48e-3) ≈ | 2.8169e-1 (1.09e-1) + | 2.4941e-1 (4.87e-2) |
| | 10 | 4.1815e-1 (1.94e-1) ≈ | 3.6840e-1 (2.06e-1) + | 1.2783e+0 (1.01e-1) + | 4.6369e-1 (4.24e-2) + | 3.5324e-1 (2.13e-2) + | 3.1320e-1 (1.14e-1) — | 3.2193e-1 (1.14e-1) |
| | 15 | 6.9167e-1 (1.11e-1) + | 6.0973e-1 (2.64e-1) ≈ | 1.4795e+0 (1.24e-1) + | 9.2356e-1 (1.62e-2) + | 2.7072e-1 (4.98e-3) — | 3.5180e-1 (1.62e-1) — | 4.5615e-1 (5.81e-2) |
| C3-DTLZ1 | 3 | 6.4693e-2 (1.20e-2) + | 5.4988e-2 (1.46e-2) ≈ | 7.0098e-2 (6.30e-3) + | 6.3010e-2 (9.09e-3) + | 6.3393e-2 (2.65e-2) + | 6.3114e-2 (4.08e-3) + | 4.5845e-2 (1.70e-3) |
| | 5 | 1.1056e-1 (6.53e-3) + | 1.1784e-1 (2.76e-3) + | 2.6861e-1 (2.61e-1) ≈ | 1.9334e-1 (3.41e-2) + | 1.2471e-1 (8.90e-4) + | 1.2161e-1 (3.38e-3) + | 1.0797e-1 (1.24e-3) |
| | 8 | 2.3654e-1 (7.48e-2) + | 2.0254e-1 (1.53e-3) — | 9.6476e-1 (3.88e-1) + | 3.2248e-1 (2.65e-2) + | 2.4484e-1 (5.66e-3) + | 2.2855e-1 (5.02e-3) + | 2.2330e-1 (7.13e-2) |
| | 10 | 3.2904e-1 (1.02e-1) + | 2.1208e-1 (3.34e-3) — | 7.2867e-1 (1.94e-1) + | 3.2983e-1 (2.48e-2) + | 2.7945e-1 (3.23e-2) ≈ | 2.3413e-1 (3.53e-3) + | 2.7966e-1 (7.98e-2) |
| | 15 | 4.1706e-1 (4.46e-2) ≈ | 2.7696e-1 (5.98e-3) — | 1.2996e+0 (7.86e-1) + | 4.7672e-1 (3.60e-1) + | 4.9344e-1 (4.03e-2) + | 3.3203e-1 (1.21e-2) — | 4.1768e-1 (2.70e-2) |
| C3-DTLZ4 | 3 | 1.6127e-1 (2.31e-1) — | 1.1056e-1 (5.65e-2) ≈ | 4.0726e-1 (3.22e-1) + | 1.1149e-1 (7.43e-2) — | 1.1255e-1 (2.82e-3) ≈ | 1.1101e-1 (2.70e-3) ≈ | 3.2507e-1 (3.45e-1) |
| | 5 | 2.6257e-1 (6.29e-2) ≈ | 2.6852e-1 (2.05e-2) + | 4.6611e-1 (9.07e-2) + | 4.3049e-1 (3.28e-2) + | 2.9092e-1 (2.56e-3) + | 2.6579e-1 (2.55e-3) + | 2.6156e-1 (6.47e-2) |
| | 8 | 7.9595e-1 (8.63e-2) ≈ | 5.1933e-1 (7.29e-3) + | 8.4917e-1 (1.52e-1) ≈ | 6.5860e-1 (2.45e-2) — | 5.4648e-1 (4.03e-3) + | 5.1464e-1 (6.35e-3) — | 7.9011e-1 (1.13e-1) |
| | 10 | 8.5162e-1 (7.27e-2) ≈ | 6.0624e-1 (2.67e-3) — | 7.6730e-1 (6.30e-2) — | 7.3312e-1 (2.27e-2) — | 6.2175e-1 (5.73e-3) — | 5.6029e-1 (1.96e-3) — | 8.1663e-1 (8.95e-2) |
| | 15 | 1.3047e+0 (1.37e-1) + | 8.0836e-1 (7.22e-4) — | 1.2760e+0 (2.96e-1) + | 8.9505e-1 (1.32e-2) — | 8.0489e-1 (5.63e-4) — | 7.7564e-1 (6.06e-3) — | 1.1067e+0 (1.23e-1) |
| DC1-DTLZ1 | 3 | 1.7701e-2 (2.22e-2) ≈ | 1.1530e-1 (7.62e-2) + | 6.1464e-1 (1.01e+0) + | 2.5552e-2 (2.78e-2) + | 1.0115e-2 (2.32e-4) ≈ | 1.5944e-2 (1.20e-3) + | 1.0579e-2 (2.48e-4) |
| | 5 | 3.5007e-2 (2.65e-4) — | 9.5875e-2 (7.72e-2) + | 3.7816e-1 (3.25e-1) + | 1.1139e-1 (1.03e-1) + | 3.5557e-2 (3.28e-4) ≈ | 4.3671e-2 (1.97e-3) + | 3.5492e-2 (7.70e-4) |
| | 8 | 1.1672e-1 (8.09e-2) ≈ | 8.7019e-2 (6.11e-2) ≈ | 8.6818e-1 (2.28e+0) + | 1.4054e-1 (6.20e-2) + | 1.3913e-1 (7.72e-2) + | 1.4428e-1 (7.11e-2) + | 6.7155e-2 (1.86e-3) |
| | 10 | 9.7234e-2 (5.14e-2) + | 1.2225e-1 (5.84e-2) + | 3.2460e-1 (2.69e-1) + | 1.4275e-1 (5.61e-2) + | 6.1893e-2 (2.13e-2) ≈ | 1.3037e-1 (1.98e-2) + | 6.9388e-2 (3.29e-2) |
| | 15 | 2.5685e-1 (7.50e-2) + | 1.7743e-1 (4.03e-2) ≈ | 5.9391e-1 (1.66e+0) + | 1.9669e-1 (4.03e-2) + | 1.4850e-1 (1.05e-2) — | 1.8399e-1 (2.13e-2) ≈ | 1.8928e-1 (3.27e-2) |
| DC1-DTLZ3 | 3 | 7.7459e-2 (8.79e-2) ≈ | 2.5210e+0 (6.76e+0) + | 2.6072e+1 (2.08e+1) + | 2.7010e-1 (5.83e-1) ≈ | 3.8855e-2 (2.68e-2) + | 2.7972e-2 (6.35e-3) — | 3.8561e-2 (3.96e-2) |
| | 5 | 1.0926e-1 (4.34e-3) — | 5.4292e-1 (1.99e-1) + | 1.8586e+1 (1.47e+1) + | 2.1450e+0 (5.74e+0) + | 2.1411e-1 (7.77e-2) + | 1.3430e-1 (8.42e-3) + | 1.1255e-1 (3.65e-3) |
| | 8 | 6.8658e-1 (2.98e-1) ≈ | 6.8627e-1 (1.75e-1) — | 6.4296e+0 (1.92e+1) + | 9.8144e-1 (2.17e+0) + | 7.4596e-1 (1.59e-1) ≈ | 4.3032e-1 (6.24e-2) — | 7.5589e-1 (2.22e-1) |
| | 10 | 5.7757e-1 (7.12e-2) + | 4.4867e-1 (1.46e-1) ≈ | 5.3989e+0 (1.21e+1) + | 6.2885e-1 (4.33e-2) + | 5.3151e-1 (1.98e-2) + | 4.1334e-1 (7.19e-3) ≈ | 4.6197e-1 (9.32e-2) |
| | 15 | 3.5472e+0 (3.35e+0) + | 6.1183e-1 (1.98e-1) — | 2.0765e+1 (4.72e+1) ≈ | 7.9473e+0 (1.81e+1) + | 8.7800e-1 (2.29e-1) — | 9.6455e-1 (1.12e-1) ≈ | 9.2587e-1 (1.27e-1) |
| DC2-DTLZ1 | 3 | Infeasible | Infeasible | Infeasible | Infeasible | 2.3392e-2 (2.94e-4) + | Infeasible | 2.0562e-2 (4.76e-5) |
| | 5 | Infeasible | Infeasible | Infeasible | Infeasible | 6.1595e-2 (2.15e-4) + | Infeasible | 5.2710e-2 (8.86e-5) |
| | 8 | Infeasible | Infeasible | Infeasible | Infeasible | 1.1132e-1 (1.33e-3) + | Infeasible | 9.9903e-2 (3.51e-3) |
| | 10 | Infeasible | Infeasible | Infeasible | Infeasible | 1.3340e-1 (1.99e-3) + | Infeasible | 1.2312e-1 (2.21e-3) |
| | 15 | Infeasible | Infeasible | Infeasible | Infeasible | 2.1108e-1 (5.81e-3) + | Infeasible | 2.0012e-1 (2.81e-3) |
| DC2-DTLZ3 | 3 | Infeasible | Infeasible | Infeasible | Infeasible | 5.4695e-2 (2.48e-4) ≈ | Infeasible | 5.4566e-2 (1.29e-4) |
| | 5 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 1.6515e-1 (1.95e-5) |
| | 8 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 3.8180e-1 (8.92e-2) |
| | 10 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 4.6032e-1 (7.18e-2) |
| | 15 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible |
| DC3-DTLZ1 | 3 | 9.4914e-2 (9.76e-2) + | 6.5415e-1 (1.53e+0) + | 7.4301e+0 (8.10e+0) + | 6.1489e-2 (1.69e-1) + | 1.0768e-2 (4.03e-4) — | 1.0796e-1 (1.43e-1) + | 1.1489e-2 (5.51e-4) |
| | 5 | 5.3751e-2 (3.31e-2) + | 1.5392e-1 (8.28e-2) + | 9.1535e+0 (1.01e+1) + | 4.1103e-1 (3.72e-1) + | 3.4773e-2 (3.90e-3) + | 3.1988e-2 (1.72e-2) + | 2.9012e-2 (1.30e-2) |
| | 8 | 9.2965e-2 (3.11e-2) + | 1.8205e-1 (9.13e-2) + | 5.1377e+0 (8.87e+0) + | 1.0667e+0 (1.22e+0) + | 1.7178e+0 (8.17e+0) + | 6.9990e-2 (2.06e-2) + | 4.8288e-2 (5.22e-3) |
| | 10 | 7.7301e-2 (2.39e-2) + | 1.3402e-1 (8.53e-2) + | 4.3745e+0 (5.39e+0) + | 2.8098e-1 (8.20e-2) + | Infeasible | 5.6680e-2 (4.06e-3) + | 4.6598e-2 (4.17e-3) |
| | 15 | 7.7899e-1 (8.81e-1) + | 3.0366e+0 (2.10e+0) + | 1.4460e+1 (9.22e+0) + | 3.6798e+0 (3.62e+0) + | Infeasible | 6.8743e-1 (6.40e-1) ≈ | 3.3576e-1 (6.97e-2) |
| DC3-DTLZ3 | 3 | 2.6568e+0 (1.56e+0) + | 3.2402e+1 (3.39e+1) + | 6.1044e+1 (3.98e+1) + | 8.1662e+0 (1.59e+1) + | 5.1671e-2 (1.75e-2) ≈ | 4.5776e+0 (2.75e+0) + | 2.7685e-1 (5.79e-1) |
| | 5 | 2.6568e+0 (1.56e+0) + | 3.2402e+1 (3.39e+1) + | 6.1044e+1 (3.98e+1) + | 8.1662e+0 (1.59e+1) + | 5.1671e-2 (1.75e-2) ≈ | 4.5776e+0 (2.75e+0) + | 2.7685e-1 (5.79e-1) |
| | 8 | 2.6568e+0 (1.56e+0) + | 3.2402e+1 (3.39e+1) + | 6.1044e+1 (3.98e+1) + | 8.1662e+0 (1.59e+1) + | 5.1671e-2 (1.75e-2) ≈ | 4.5776e+0 (2.75e+0) + | 2.7685e-1 (5.79e-1) |
| | 10 | 7.4588e+0 (3.06e+0) + | 1.1923e+0 (3.71e-1) ≈ | 1.8891e+0 (2.81e+0) ≈ | 5.4020e+1 (4.52e+1) + | 3.7178e-1 (7.65e-2) — | 5.2769e-1 (1.57e-1) — | 1.4573e+0 (1.54e+0) |
| | 15 | 1.3689e+1 (1.07e+1) + | 2.0934e+1 (9.74e+0) + | 1.5852e+1 (1.89e+1) + | 8.6460e+1 (5.73e+1) + | 2.1973e+0 (2.01e+0) — | 4.3974e+0 (4.66e+0) ≈ | 2.3841e+0 (1.43e+0) |

“Infeasible” stands for the algorithm cannot find feasible solutions in all 30 runs.

TABLE S-IV
COMPARATIVE RESULTS OF HV VALUES (MEAN AND STANDARD DEVIATION) OBTAINED BY A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AND, AND DCNSGA-III, RESPECTIVELY.

| Problem | m | A-NSGA-III | C-MOEA/D | I-DBEA | PPS | C-TAEA | C-AND | DCNSGA-III |
|-----------|----|--------------------------------|--------------------------------|---------------------|--------------------------------|---------------------------------|--------------------------------|-------------------------------|
| CF8 | 3 | Infeasible | Infeasible | Infeasible | 4.2406e-1(1.09e-2) | 1.9378e-1(6.51e-2)+ | Infeasible | 2.6138e-1(6.48e-2) |
| CF9 | 3 | 3.5940e-1(8.84e-2)≈ | 4.3778e-1(1.75e-2)− | 8.8064e-2(4.98e-2)+ | 5.0806e-1(1.63e-2) | 3.9182e-1(1.56e-2)≈ | 3.8477e-1(2.62e-2)≈ | 3.5405e-1(8.00e-2) |
| CF10 | 3 | Infeasible | Infeasible | Infeasible | 2.7805e-1(7.12e-2) | 1.0265e-1(3.36e-2)+ | Infeasible | 1.9366e-1(6.74e-2) |
| MW4 | 3 | 8.3825e-1(3.98e-3)+ | 8.1085e-1(8.36e-3)+ | 1.9614e-1(9.33e-2)+ | Infeasible | 8.4074e-1(2.52e-4) | 8.3214e-1(1.15e-2)+ | 8.4033e-1(7.85e-4) |
| MW8 | 3 | 5.0368e-1(6.21e-2)≈ | 5.2983e-1(1.03e-2)≈ | 5.1517e-2(3.09e-2)+ | 3.5554e-1(9.95e-2)+ | 5.3083e-1(1.25e-2)≈ | 4.1824e-1(1.00e-1)+ | 5.3015e-1(9.53e-3) |
| MW14 | 3 | 4.6162e-1(1.13e-2)≈ | 4.1485e-1(6.76e-3)+ | 1.2004e-2(7.25e-3)+ | 4.1615e-1(2.15e-2)+ | 4.6793e-1(5.20e-3) | 3.0300e-1(8.36e-2)+ | 4.6420e-1(2.34e-3) |
| DASCMP7 | 3 | 2.8498e-1(1.22e-3)− | 2.4667e-1(6.44e-2)≈ | 1.3813e-2(3.95e-2)+ | 2.5942e-1(3.68e-2)+ | 2.8504e-1(5.34e-3)− | 2.8547e-1(8.26e-3) | 2.8088e-1(2.26e-3) |
| DASCMP8 | 3 | 1.9050e-1(3.09e-2)+ | 1.6520e-1(5.85e-2)+ | 6.4248e-3(1.50e-2)+ | 1.8102e-1(1.68e-2)+ | 2.0234e-1(3.75e-3)≈ | 2.0610e-1(1.74e-3) | 2.0404e-1(7.56e-4) |
| DASCMP9 | 3 | 1.1957e-1(7.78e-3)≈ | 1.9951e-1(8.83e-4)− | 4.2268e-2(5.27e-3)+ | 2.0636e-1(1.70e-3) | 1.5702e-1(1.51e-2)− | 1.0660e-1(2.75e-2)≈ | 1.0852e-1(2.30e-2) |
| C1-DTLZ1 | 3 | 8.2037e-1(1.44e-2)+ | 8.3111e-1(2.77e-3)+ | 9.5747e-2(7.95e-2)+ | 7.9241e-1(3.07e-2)+ | 8.3039e-1(1.21e-2)+ | 8.2947e-1(5.30e-3)+ | 8.3322e-1(7.53e-3) |
| | 5 | 9.6765e-1(1.25e-2)+ | 9.6698e-1(2.02e-3)+ | 1.1844e-1(1.16e-1)+ | Infeasible | 9.7679e-1(1.20e-3)≈ | 9.6519e-1(1.53e-2)≈ | 9.7508e-1(4.66e-3) |
| | 8 | 9.6876e-1(2.37e-2)+ | 9.7730e-1(2.85e-3)+ | 1.2084e-1(1.02e-1)+ | 9.7384e-1(3.23e-2)≈ | 9.9566e-1(1.41e-3) | 9.6808e-1(5.36e-2)≈ | 9.8179e-1(1.15e-2) |
| | 10 | 9.7799e-1(1.92e-2)+ | 9.8479e-1(1.92e-3)+ | 1.3506e-1(1.37e-1)+ | 9.9513e-1(1.97e-3)≈ | 9.9791e-1(3.65e-3) | 9.8287e-1(1.83e-2)≈ | 9.9194e-1(6.94e-3) |
| | 15 | 9.6990e-1(3.04e-2)+ | 9.8176e-1(5.00e-3)+ | 6.1801e-2(4.25e-2)+ | 9.6221e-1(5.27e-2)+ | 9.9576e-1(7.66e-3)≈ | 9.8652e-1(1.05e-2)+ | 9.9684e-1(5.02e-3) |
| C1-DTLZ3 | 3 | 3.3338e-1(2.59e-1)≈ | 3.0303e-1(2.16e-1)≈ | 1.1294e-1(2.21e-1)+ | 3.9410e-1(2.47e-1)≈ | 2.8686e-1(2.73e-1)≈ | 2.7928e-2(1.89e-1)+ | 3.1273e-1(2.78e-1) |
| | 5 | 8.0076e-1(9.38e-3)+ | 6.0344e-1(1.67e-1)+ | 8.1139e-1(1.09e-3)≈ | 4.1268e-1(2.16e-1)+ | 6.6086e-1(4.26e-2)+ | 8.0234e-1(4.09e-3)+ | 8.1194e-1(7.80e-4) |
| | 8 | 7.5165e-1(1.21e-1)+ | 6.1442e-1(1.76e-1)+ | 9.7072e-2(3.39e-2)+ | 5.0676e-1(5.78e-2)+ | 7.9914e-1(2.60e-2)+ | 7.0275e-1(2.02e-1)+ | 8.8632e-1(7.96e-2) |
| | 10 | 8.2564e-1(5.83e-2)+ | 4.8434e-1(1.62e-2)+ | 9.0907e-2(3.18e-6)+ | 4.4896e-1(6.28e-2)+ | 9.0948e-1(1.27e-2)+ | 8.7843e-1(5.44e-2)+ | 9.4101e-1(5.12e-2) |
| | 15 | 1.0945e-1(1.87e-1)+ | 9.4066e-1(1.36e-2) | 9.0897e-2(1.73e-5)+ | 1.7318e-1(5.91e-2)+ | 7.8556e-1(2.55e-1)≈ | 1.7984e-1(1.45e-1)+ | 8.3910e-1(7.22e-2) |
| C2-DTLZ2 | 3 | 5.0763e-1(3.32e-3) | 4.7208e-1(7.16e-3)+ | 3.3824e-2(3.43e-2)+ | 5.0117e-1(2.78e-3)≈ | 5.0410e-1(2.82e-3)≈ | 5.0389e-1(5.74e-2)− | 5.0171e-1(2.11e-3) |
| | 5 | 7.4258e-1(3.22e-2)+ | 6.7281e-1(7.27e-3)+ | 3.1005e-2(3.72e-2)+ | 5.5906e-1(6.01e-2)+ | 7.4057e-1(2.06e-3)+ | 7.6111e-1(1.25e-3) | 7.4693e-1(2.69e-3) |
| | 8 | 7.4129e-1(1.91e-1)≈ | 5.1812e-1(1.95e-1)+ | 4.8087e-2(5.35e-2)+ | 5.3041e-1(6.68e-2)+ | 7.9526e-1(8.04e-3)+ | 8.1807e-1(1.17e-1)≈ | 8.2979e-1(2.76e-2) |
| | 10 | 7.9787e-1(1.73e-1)+ | 7.3422e-1(2.12e-1)+ | 5.9943e-2(3.85e-2)+ | 6.1815e-1(4.39e-2)+ | 8.7485e-1(3.83e-3) | 8.6816e-1(1.26e-1)≈ | 8.7289e-1(4.64e-2) |
| | 15 | 7.0308e-1(1.41e-1)+ | 6.3875e-1(3.13e-1)+ | 1.3028e-2(2.64e-2)+ | 2.7349e-1(1.51e-1)+ | 9.3182e-1(4.06e-3) | 8.7940e-1(1.84e-1)+ | 8.8257e-1(3.08e-2) |
| C3-DTLZ1 | 3 | 8.4744e-1(1.17e-2)+ | 8.5779e-1(1.35e-2)+ | 8.4416e-1(6.35e-3)+ | 8.3818e-1(1.35e-2)+ | 8.5034e-1(2.97e-2)+ | 8.4763e-1(5.06e-3)+ | 8.6746e-1(2.17e-3) |
| | 5 | 9.7883e-1(1.35e-3)≈ | 9.7555e-1(1.02e-3)+ | 8.0668e-1(2.88e-1)≈ | 9.2596e-1(5.08e-2)+ | 9.7702e-1(2.40e-4)+ | 9.7332e-1(1.35e-3)+ | 9.7936e-1(4.43e-4) |
| | 8 | 9.8330e-1(4.69e-2)+ | 9.9577e-1(3.28e-4)− | 2.8764e-1(2.70e-1)+ | 9.2786e-1(2.97e-2)+ | 9.9628e-1(8.94e-4) | 9.9450e-1(8.32e-4)− | 9.8670e-1(3.31e-2) |
| | 10 | 9.7089e-1(3.65e-2)+ | 9.9938e-1(7.73e-5)≈ | 5.5569e-1(2.16e-1)+ | 9.3357e-1(2.86e-2)+ | 9.9868e-1(3.03e-3)≈ | 9.9903e-1(1.77e-4)≈ | 9.8872e-1(1.63e-2) |
| | 15 | 9.8386e-1(1.59e-2)≈ | 9.9963e-1(1.45e-4) | 2.2155e-1(2.40e-1)+ | 7.4041e-1(2.78e-1)+ | 9.6084e-1(4.27e-2)+ | 9.9746e-1(1.34e-3)− | 9.9208e-1(4.22e-3) |
| C3-DTLZ4 | 3 | 7.6642e-1(1.89e-2)− | 7.8329e-1(1.71e-2)≈ | 6.1919e-1(1.32e-1)+ | 7.9165e-1(2.66e-2) | 7.8470e-1(1.50e-3)≈ | 7.8355e-1(2.34e-3)≈ | 7.0967e-1(1.19e-1) |
| | 5 | 9.6016e-1(1.03e-2)≈ | 9.6002e-1(1.21e-3)− | 8.8700e-1(2.56e-2)+ | 9.3693e-1(6.04e-3)+ | 9.5712e-1(6.44e-4)+ | 9.5892e-1(8.97e-4)+ | 9.5995e-1(1.10e-2) |
| | 8 | 9.6240e-1(1.82e-2)≈ | 9.9493e-1(2.46e-4) | 9.2386e-1(5.04e-2)+ | 9.8054e-1(2.93e-3)− | 9.9439e-1(1.50e-4)− | 9.9465e-1(2.93e-4)− | 9.6102e-1(2.18e-2) |
| | 10 | 9.8517e-1(9.13e-3)≈ | 9.9934e-1(3.72e-5) | 9.8846e-1(4.46e-3)≈ | 9.9156e-1(1.59e-3)≈ | 9.9903e-1(4.61e-5)− | 9.9921e-1(4.24e-5)− | 9.8755e-1(1.02e-2) |
| | 15 | 9.3349e-1(3.99e-2)+ | 9.9996e-1(7.28e-6)− | 8.1914e-1(2.56e-1)+ | 9.9722e-1(7.21e-4)− | 9.9997e-1(5.20e-6) | 9.9991e-1(1.45e-5)− | 9.8393e-1(1.78e-2) |
| DC1-DTLZ1 | 3 | 6.3755e-1(3.15e-2)+ | 4.8228e-1(9.98e-2)+ | 1.6766e-1(1.83e-1)+ | 6.0783e-1(5.28e-2)+ | 6.4899e-1(1.11e-3)≈ | 6.2964e-1(5.37e-3)+ | 6.4935e-1(1.40e-3) |
| | 5 | 8.1004e-1(2.22e-4)+ | 7.0701e-1(1.73e-1)+ | 2.7291e-1(2.52e-1)+ | 6.6295e-1(2.05e-1)+ | 8.1108e-1(2.42e-4) | 7.9934e-1(2.05e-3)+ | 8.1029e-1(2.06e-4) |
| | 8 | 6.2109e-1(5.76e-2)≈ | 5.1949e-1(5.99e-2)+ | 1.5709e-1(1.19e-1)+ | 3.4641e-1(6.86e-2)+ | 5.7134e-1(6.76e-2)+ | 5.3984e-1(5.70e-2)+ | 6.2786e-1(7.89e-3) |
| | 10 | 8.3343e-1(1.46e-2)+ | 7.4883e-1(1.40e-1)+ | 5.9205e-1(2.85e-1)+ | 8.1798e-1(6.55e-2)+ | 8.3601e-1(5.90e-5) | 8.3432e-1(4.15e-4)− | 8.3411e-1(7.29e-3) |
| | 15 | 1.7451e-1(2.52e-1)≈ | 1.6984e-1(2.14e-1)≈ | 7.3000e-2(9.37e-2)+ | 5.2633e-1(8.62e-2) | 3.0352e-1(1.17e-1)− | 4.3158e-2(4.04e-2)+ | 1.6245e-1(1.39e-1) |
| DC1-DTLZ3 | 3 | 4.4640e-1(4.25e-2)≈ | 2.4284e-1(1.23e-1)+ | 3.3604e-1(1.84e-2)+ | 3.6841e-1(1.55e-1)≈ | 4.6891e-1(1.60e-2)− | 4.8035e-1(4.16e-3) | 4.6281e-1(1.59e-2) |
| | 5 | 7.8341e-1(1.39e-3) | 5.9135e-1(1.72e-1)+ | 5.7259e-3(1.92e-2)+ | 2.1250e-1(2.56e-1)+ | 6.8475e-1(4.28e-2)+ | 7.7943e-1(2.53e-3)+ | 7.8236e-1(1.88e-3) |
| | 8 | 7.6723e-1(1.23e-1)≈ | 7.7455e-1(1.73e-1)− | 4.8599e-2(4.42e-2)+ | 4.3862e-1(1.23e-1)+ | 7.5935e-1(3.02e-2)≈ | 8.9110e-1(1.23e-2) | 7.4255e-1(9.23e-2) |
| | 10 | 8.2849e-1(6.15e-2)+ | 8.6170e-1(1.85e-1)≈ | 5.6151e-2(3.85e-2)+ | 4.6553e-1(8.65e-2)+ | 8.8374e-1(1.55e-2)≈ | 9.4726e-1(5.59e-3)≈ | 9.0438e-1(6.62e-2) |
| | 15 | 2.5670e-2(5.96e-2)+ | 5.5659e-1(9.29e-2) | 4.7228e-2(3.52e-2)+ | 1.9069e-2(3.34e-2)+ | 2.4744e-1(1.07e-1)+ | 1.6951e-1(1.01e-1)+ | 3.1949e-1(1.14e-1) |
| DC2-DTLZ1 | 3 | Infeasible | Infeasible | Infeasible | Infeasible | 8.3899e-1(4.56e-4)+ | Infeasible | 8.4165e-1(7.34e-4) |
| | 5 | Infeasible | Infeasible | Infeasible | Infeasible | 9.7691e-1(1.66e-4)+ | Infeasible | 9.7990e-1(1.99e-4) |
| | 8 | Infeasible | Infeasible | Infeasible | Infeasible | 9.9699e-1(4.36e-4)+ | Infeasible | 9.9716e-1(9.01e-4) |
| | 10 | Infeasible | Infeasible | Infeasible | Infeasible | 9.9941e-1(7.59e-5)+ | Infeasible | 9.9969e-1(9.67e-5) |
| | 15 | Infeasible | Infeasible | Infeasible | Infeasible | 9.9979e-1(3.09e-5) | Infeasible | 9.9906e-1(3.44e-4) |
| DC2-DTLZ3 | 3 | Infeasible | Infeasible | Infeasible | Infeasible | 5.3491e-1(6.74e-2)≈ | Infeasible | 5.3576e-1(2.31e-2) |
| | 5 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 7.9899e-1(1.54e-3) |
| | 8 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 8.1380e-1(5.90e-2) |
| | 10 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | 9.0094e-1(5.31e-2) |
| | 15 | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible | Infeasible |
| DC3-DTLZ1 | 3 | 4.3827e-1(2.17e-1)+ | 1.6143e-1(1.65e-1)+ | 0.0000e+0(0.00e+0)+ | 5.9171e-1(1.63e-1)+ | 6.5840e-1(9.94e-4)+ | 4.4886e-1(2.34e-1)+ | 6.6126e-1(4.08e-3) |
| | 5 | 6.8732e-1(8.92e-2)≈ | 4.8293e-1(2.01e-1)+ | 5.8107e-3(3.18e-2)+ | 2.5609e-1(2.13e-1)+ | 7.3385e-1(7.13e-3)+ | 7.4800e-1(3.21e-2)≈ | 7.5480e-1(5.70e-3) |
| | 8 | 5.9264e-1(1.13e-1)+ | 3.7597e-1(1.48e-1)+ | 5.2404e-3(2.02e-2)+ | 1.5454e-1(1.76e-1)+ | 3.5665e-1(1.61e-1)+ | 7.1013e-1(7.14e-2) | 6.9623e-1(2.07e-2) |
| | 10 | 6.1449e-1(1.12e-1)+ | 3.7781e-1(1.55e-1)+ | 6.3148e-3(2.48e-2)+ | 3.2666e-1(1.74e-1)+ | Infeasible | 7.4425e-1(5.13e-3) | 7.0015e-1(1.85e-2) |
| | 15 | 1.9004e-1(1.30e-1)+ | 4.0722e-2(8.53e-2)+ | 0.0000e+0(0.00e+0)+ | 5.7453e-2(9.89e-2)+ | Infeasible | 2.0784e-1(1.55e-1)≈ | 2.8513e-1(7.64e-2) |
| DC3-DTLZ3 | 3 | 3.0870e-3(1.29e-2)+ | 1.8764e-2(8.36e-2)+ | 0.0000e+0(0.00e+0)+ | 2.3058e-1(1.96e-1)+ | 4.1582e-1(2.46e-2) ≈ | 0.0000e+0(0.00e+0)+ | 3.5079e-1(1.73e-1) |
| | 5 | 2.7257e-1(2.34e-1)+ | 4.0867e-2(1.42e-1)+ | 0.0000e+0(0.00e+0)+ | 2.6815e-2(8.59e-2)+ | 6.7709e-1(8.16e-3)+ | 4.1649e-1(2.49e-1)+ | 7.2092e-1(3.29e-3) |
| | 8 | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0) |
| | 10 | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0) |
| | 15 | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 0.0000e+0(0.00e+0)≈ | 1.5777e-6(7.12e-6) | 0.0000e+0(0.00e+0) |

“Infeasible” stands for the algorithm cannot find feasible solutions in all 30 runs.

TABLE S-II
NUMBER OF MAXIMUM FUNCTION EVALUATIONS FOR DIFFERENT
TEST PROBLEMS.

| Problem | m | n | FEs |
|-----------------------------------|-----|-----|-----------|
| CF8-10 | 3 | 10 | 150,000 |
| MW4, MW8, MW14 | 3 | 15 | 60,000 |
| DAS-CMOP7-9 | 3 | 30 | 300,000 |
| C1-DTLZ1 | 3 | 7 | 46,000 |
| | 5 | 9 | 127,200 |
| | 8 | 12 | 124,800 |
| | 10 | 14 | 276,000 |
| | 15 | 19 | 204,000 |
| C1-DTLZ3 | 3 | 12 | 92,000 |
| | 5 | 14 | 318,000 |
| | 8 | 17 | 390,000 |
| | 10 | 19 | 966,000 |
| | 15 | 24 | 680,000 |
| C2-DTLZ2 | 3 | 12 | 23,000 |
| | 5 | 14 | 74,200 |
| | 8 | 17 | 78,000 |
| | 10 | 19 | 207,000 |
| | 15 | 24 | 136,000 |
| C3-DTLZ1, DC1-DTLZ1, DC3-DTLZ1 | 3 | 7 | 69,000 |
| | 5 | 9 | 265,000 |
| | 8 | 12 | 312,000 |
| | 10 | 14 | 828,000 |
| | 15 | 19 | 544,000 |
| C3-DTLZ4, DC1-DTLZ3, DC3-DTLZ3 | 3 | 12 | 69,000 |
| | 5 | 14 | 265,000 |
| | 8 | 17 | 312,000 |
| | 10 | 19 | 828,000 |
| | 15 | 24 | 544,000 |
| DC2-DTLZ1 | 3 | 7 | 138,000 |
| | 5 | 9 | 530,000 |
| | 8 | 12 | 624,000 |
| | 10 | 14 | 1,656,000 |
| | 15 | 19 | 1,088,000 |
| DC2-DTLZ3 | 3 | 12 | 138,000 |
| | 5 | 14 | 530,000 |
| | 8 | 17 | 624,000 |
| | 10 | 19 | 1,656,000 |
| | 15 | 24 | 1,088,000 |

percentage of runs where at least one feasible solution was found. We employ CF10 and DC2-DTLZ3 as test problems.

Fig. S-1 shows the feasibility ratio of three mating selection operators on CF10 and DC2-DTLZ3, respectively. From Fig. S-1, we can clearly see that *Variant3* has the worst performance among these three operators. The reason is that it neglects the constraint information, which cannot guarantee to generate enough solutions with small constraint violations. Particularly at the latter

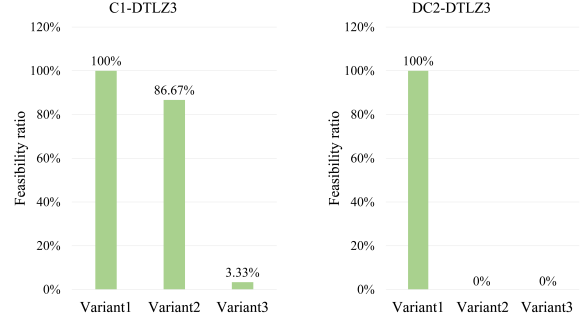


Fig. S-1. Comparison of feasibility ratio of three different mating selection operators on CF10 problem and DC2-DTLZ3 problem, respectively.

stage of evolution, the poor tournament selection cannot create enough offsprings with a low degree of constraint violation to push the population towards the feasible area. *Variant2* performs better than *Variant3*, but it still cannot find feasible solutions consistently on CF10 and DC2-DTLZ3 problems. This is because excessively emphasize feasibility and ignore the information of objectives during the offspring generation process can lead to the population stagnate in infeasible regions easily. By contrast, the proposed mating selection operator (*Variant1*) can find feasible solutions in all runs. A key to the proposed algorithm is to generate more ε -feasible solutions. On the one hand, the majority of ε -feasible solutions assure that the population can focus on searching for well-balanced solutions between convergence and diversity, which is beneficial to the population to cross the large and separate infeasible local optima. On the other hand, the dynamically reduced ε constraint boundary ensures these ε -feasible solutions towards the feasible region.

S-IV. PARAMETER ANALYSIS

The shrink of the dynamic ε constraint boundary adopts the exponential function of the simulated annealing algorithm. It has a crux parameter cp , which controls the decreasing trend of the dynamic ε constraint boundary. In this section, the sensitivity analysis of cp is conducted as follows.

Fig. S-2 plots the changes of the dynamic ε constraint boundary with different cp values. It can be observed that the dynamic ε constraint boundary shrinks very fast with $cp=1$. When the value of cp increases, the shrinkage speed of the dynamic ε constraint boundary also slows down.

Figs. S-3-S-5 plot the performance, in terms of IGD, of DCNSGA-III with different cp values (1-10) on C1-DTLZ1, C1-DTLZ3, and DC3-DTLZ1 over 30 independent runs, respectively. It can be observed that the optimal value of cp is problem-dependent. To elaborate,

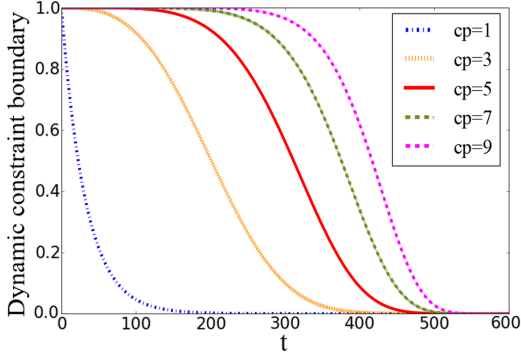


Fig. S-2. Changes of the dynamic ε constraint boundary with different cp values ($\varepsilon^{(0)} = 1$ and $T=600$).

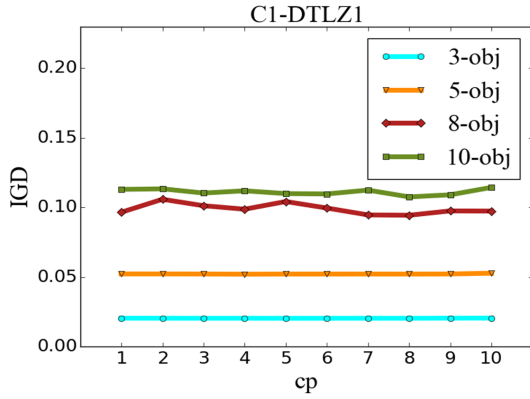


Fig. S-3. Mean IGD values obtained by DCNSGA-III with different values of cp on C1-DTLZ1 test problem over 30 runs.

the performance on the C1-DTLZ1 test problem in Fig. S-3 is not sensitive to the setting of cp . Except for the C1-DTLZ3 problem with 3 objectives, the performance of the C1-DTLZ3 problem with 5, 8, and 10 objectives does not fluctuate greatly for different cp settings. For the C1-DTLZ3 problem with 3 objectives and DC3-DTLZ1 test problem, the results in Fig. S-4 and Fig. S-5 suggest that DCNSGA-III has the poor performance with $cp = 1, 2, 3$. The reason can be attributed that the dynamic ε constraint boundary shrinks very fast when cp has a small value, which results in the poor exploration ability in the early stage of evolution.

However, in general, DCNSGA-III is robust with regard to $cp = 4, 5, 6, 7, 8, 9, 10$. Therefore, cp is in the range of 4 to 10 could be the better choice for most problems.

S-V. PERFORMANCE ON CONSTRAINED BI-OBJECTIVE OPTIMIZATION PROBLEMS

Although the proposed algorithm is designed for constrained many-objective optimization problems, in this section, we compare the proposed DCNSGA-III with its competitors on two set of bi-objective optimization

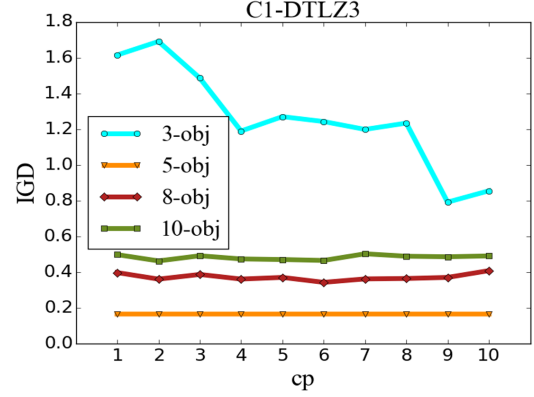


Fig. S-4. Mean IGD values obtained by DCNSGA-III with different values of cp on C1-DTLZ3 test problem over 30 runs.

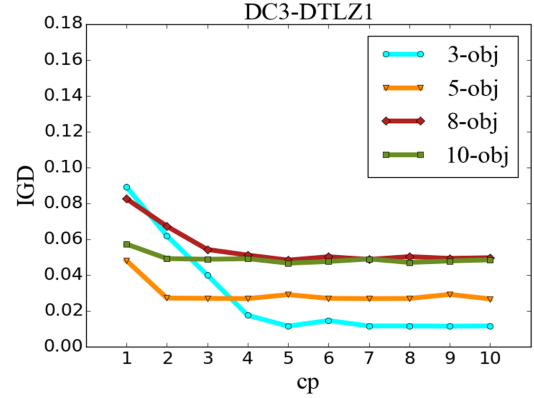


Fig. S-5. Mean IGD values obtained by DCNSGA-III with different values of cp on DC3-DTLZ1 test problem over 30 runs.

problems: C-DTLZ test suite [4] and DC-DTLZ test suite [5]. The constraint function parameters of these two-objective test problems were set as follows:

- For C1-DTLZ3: $r=6$;
- For C2-DTLZ2: $r=0.1$;
- For DC1-DTLZ1 and DC1-DTLZ3 problems: $a=3$, $b=0.5$;
- For DC2-DTLZ1 and DC2-DTLZ3 problems: $a=3$, $b=0.9$;
- For DC3-DTLZ1 and DC3-DTLZ3 problems: $a=5$, $b=0.5$.

DCNSGA-III was compared with two MOEA/D variants (C-MOEA/D [4], MOEA/D-DAE [11]) and a NSGA-III variant (C-NSGA-III [4]). The population size and the maximal number of function evaluations are set to 100 and 60,000, respectively.

Table S-V presents both the average and standard deviation of the IGD values over 30 independent runs for the four compared algorithms, where the best average and standard deviation among four algorithms are highlighted with a gray background.

TABLE S-V
COMPARATIVE RESULTS OF IGD VALUES (MEAN AND STANDARD DEVIATION) ON CONSTRAINED BI-OBJECTIVE OPTIMIZATION PROBLEMS OBTAINED BY C-NSGA-III, C-MOEA/D, MOEA/D-DAE, AND DCNSGA-III, RESPECTIVELY.

| Problem | m | C-NSGA-III | C-MOEA/D | MOEA/D-DAE | DCNSGA-III |
|-----------|-----|--------------------|--------------------|---------------------|--------------------|
| C1-DTLZ1 | 2 | 1.8353e-3(7.24e-5) | 1.7824e-3(1.06e-6) | 1.8232e-3(1.20e-5) | 1.8251e-3(5.57e-5) |
| C1-DTLZ3 | 2 | 3.0009e+0(2.01e-3) | 2.5551e+0(3.42e+0) | 4.0296e-2(1.81e-1) | 3.0007e+0(9.91e-4) |
| C2-DTLZ2 | 2 | 1.3869e-1(9.86e-2) | 3.3352e-1(1.54e-1) | 1.1837e-3(2.90e-5) | 1.7711e-3(8.45e-5) |
| C3-DTLZ1 | 2 | 4.5938e-3(9.37e-4) | 8.2651e-3(2.37e-2) | 4.5747e-3(3.02e-4) | 7.6047e-3(2.97e-3) |
| C3-DTLZ4 | 2 | 9.2471e-3(5.48e-4) | 9.6789e-3(4.59e-4) | 8.7021e-3(2.73e-4) | 1.2063e-2(1.04e-3) |
| DC1-DTLZ1 | 2 | 4.9581e-4(8.97e-5) | 1.6595e-1(7.90e-2) | 6.3965e-2(1.00e-1) | 4.8899e-4(1.34e-4) |
| DC1-DTLZ3 | 2 | 2.4300e-2(6.97e-2) | 4.4291e-1(3.18e-1) | 2.6283e-1(2.96e-1) | 3.9511e-2(8.69e-2) |
| DC2-DTLZ1 | 2 | Infeasible | Infeasible | 1.18268e-3(2.96e-5) | 1.8224e-3(5.73e-5) |
| DC2-DTLZ3 | 2 | Infeasible | Infeasible | Infeasible | Infeasible |
| DC3-DTLZ1 | 2 | 7.4914e-2(6.14e-2) | 3.1199e-1(5.01e-1) | 4.6977e-1(1.11e-1) | 1.0533e-3(1.10e-4) |
| DC3-DTLZ4 | 2 | 6.3767e-1(1.78e-1) | 3.5385e+0(2.70e+0) | 1.5661e+1(1.81e+1) | 4.9956e-1(2.08e-1) |

“Infeasible” stands for the algorithm cannot find feasible solutions in all 30 runs.

Overall, MOEA/D-DAE performs best on the two-objective C-DTLZ test suite. For MOEA/D-DAE, if a feasible subarea is found, it will continue to search other areas which could dominate the current found feasible region, so that it can cross the insurmountable infeasible area to converge to the PF on C1-DTLZ3 problem. Furthermore, if the population gets stuck in local optima of the constraint violation, the diversity enhancement scheme will be triggered to jump out of this local region, which is suitable for solving problems with multi-modal of the constraint violation.

The proposed DCNSGA-III has the best performance on the bi-objective DC-DTLZ test suite. The constraints of DC1-DTLZ and DC3-DTLZ test instances split the feasible region into a couple of narrow tapered strips, the population has the risk of being trapped in a local feasible area and all feasible areas cannot easily be found. For DCNSGA-III, at first, the population gets to its approximated unconstrained PF. In the middle and late stages of the evolution, some feasible solutions keep unchanged due to some parts of the unconstrained PF are segments of the constrained PF. Other infeasible solutions move back to the segments of the constrained PF. That is why DCNSGA-III can perform best on such type of problem.

S-VI. PLOTS OF THE OBTAINED APPROXIMATED SOLUTIONS

This section presents the plots of the final obtained approximated solutions obtained by DCNSGA-III and the six peer competitors on different test instances for 3-objective problems. Here the plots of the final obtained solutions are the one having the median IGD value. Note that the final obtained solutions may include feasible and infeasible solutions, due to an algorithm on some test problems may not obtain feasible solutions.

S-VII. THE SETTING OF FIVE REPRESENTATIVE CONSTRAINT-HANDLING TECHNIQUES

In Section VI-C of the original paper, five well-known constraint-handling techniques: CDP [12], SR [13], self-adaptive penalty (SP) [14], ε method [15], and adaptive trade-off model (ATM) [16] were embedded into the same framework of NSGA-III, namely CDP-NSGA-III, SR-NSGA-III, SP-NSGA-III, ε -NSGA-III, and ATM-NSGA-III, respectively. Some of them have parameters need to be set. The simulated binary crossover (SBX) [12] and polynomial mutation (PM) [12] were employed as reproduction operators for these five algorithms and the proposed DCNSGA-III:

- SBX: crossover probability $p_c=0.9$ and distribution index $\eta_c=30$;
- PM: mutation probability $p_m=1/n$ and distribution index $\eta_m=20$.

For ε -NSGA-III, as suggested in [17], $\theta = 0.1N$, $T_c = 0.6 \times Max_{gen}$, where Max_{gen} is the maximum number of generations, and $cp = (-5 - \log \varepsilon_0) / \log(0.05)$. For SR-NSGA-III, the probability $P_f=0.45$.

The source code of DCNSGA-III is available from <https://ruwangjiao.github.io/>.

REFERENCES

- [1] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the cec 2009 special session and competition," *University of Essex, and Nanyang technological University, technical report*, vol. 264, 2008.
- [2] Z. Ma and Y. Wang, "Evolutionary constrained multiobjective optimization: Test suite construction and performance comparisons," *IEEE Transactions on Evolutionary Computation*, 2019, doi:10.1109/TEVC.2019.2896967.
- [3] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Difficulty adjustable and scalable constrained multi-objective test problem toolkit," *Evolutionary Computation*, pp. 1–28, 2019.
- [4] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part ii: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.
- [5] K. Li, R. Chen, G. Fu, and X. Yao, "Two-archive evolutionary algorithm for constrained multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 2, pp. 303–315, 2018.
- [6] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 445–460, 2014.
- [7] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, and E. Goodman, "Push and pull search for solving constrained multi-objective optimization problems," *Swarm and Evolutionary Computation*, vol. 44, pp. 665–679, 2019.
- [8] Z.-Z. Liu, Y. Wang, and P.-Q. Huang, "And: A many-objective evolutionary algorithm with angle-based selection and shift-based density estimation," *Information Sciences*, 2018.
- [9] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 2, pp. 264–283, 2015.
- [10] Q. Zhu, Q. Lin, J. Li, C. A. C. Coello, Z. Ming, J. Chen, and J. Zhang, "An elite gene guided reproduction operator for many-objective optimization," *IEEE Transactions on Cybernetics*, pp. 1–14, 2019.
- [11] Q. Zhu, Q. Zhang, and Q. Lin, "A constrained multiobjective evolutionary algorithm with detect-and-escape strategy," *IEEE Transactions on Evolutionary Computation*, pp. 1–10, 2020, doi:10.1109/TEVC.2020.2981949.
- [12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [13] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.
- [14] W. Y. Gebre, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 514–525, 2009.
- [15] T. Takahama and S. Sakai, "Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites," in *2006 IEEE International Conference on Evolutionary Computation*. IEEE, 2006, pp. 1–8.
- [16] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 80–92, 2008.
- [17] Z. Yang, X. Cai, and Z. Fan, "Epsilon constrained method for constrained multiobjective optimization problems: some preliminary results," in *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*. ACM, 2014, pp. 1181–1186.

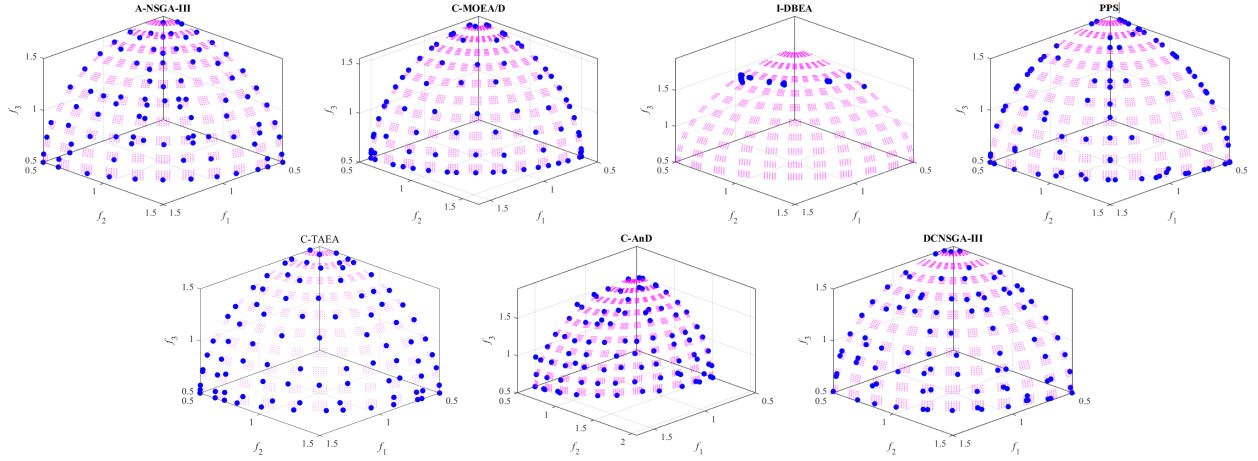


Fig. S-6. Obtained solutions for DAS-CMOP8 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

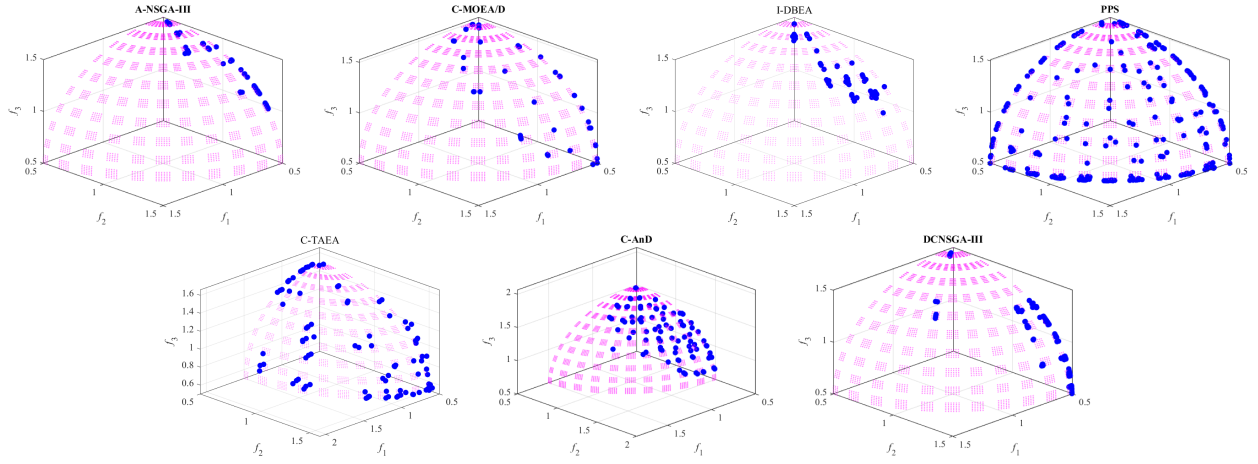


Fig. S-7. Obtained solutions for DAS-CMOP9 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

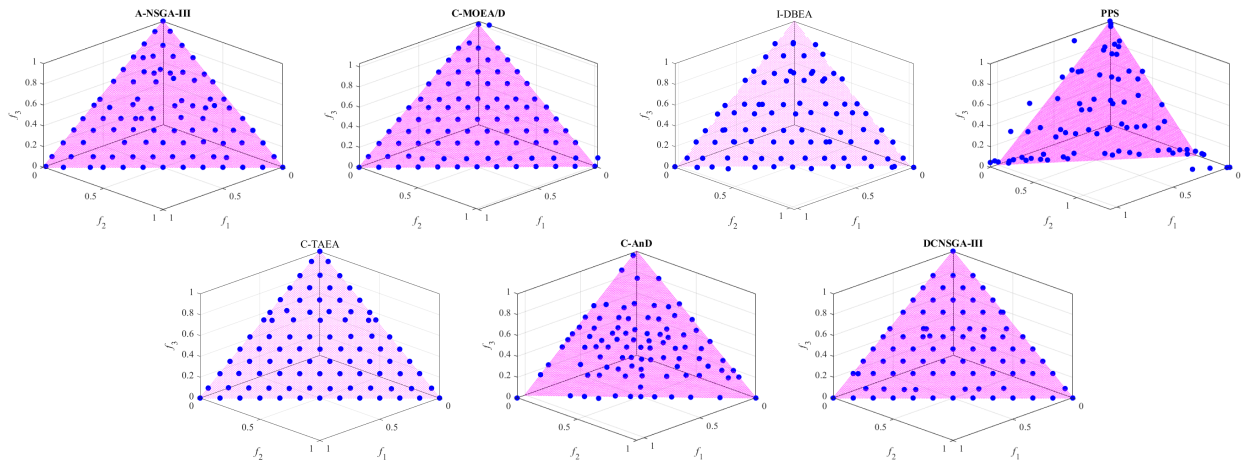


Fig. S-8. Obtained solutions for MW4 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

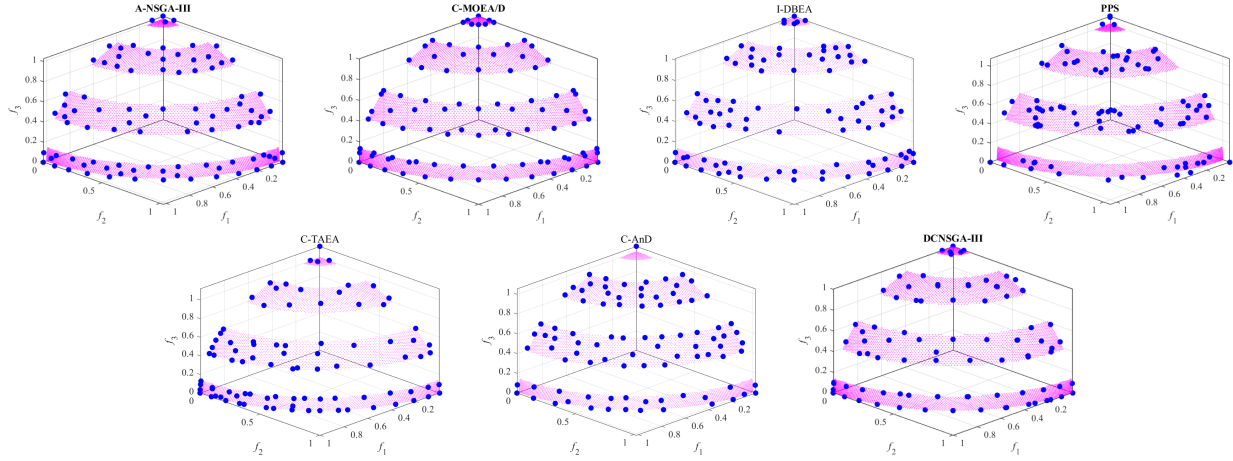


Fig. S-9. Obtained solutions for MW8 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

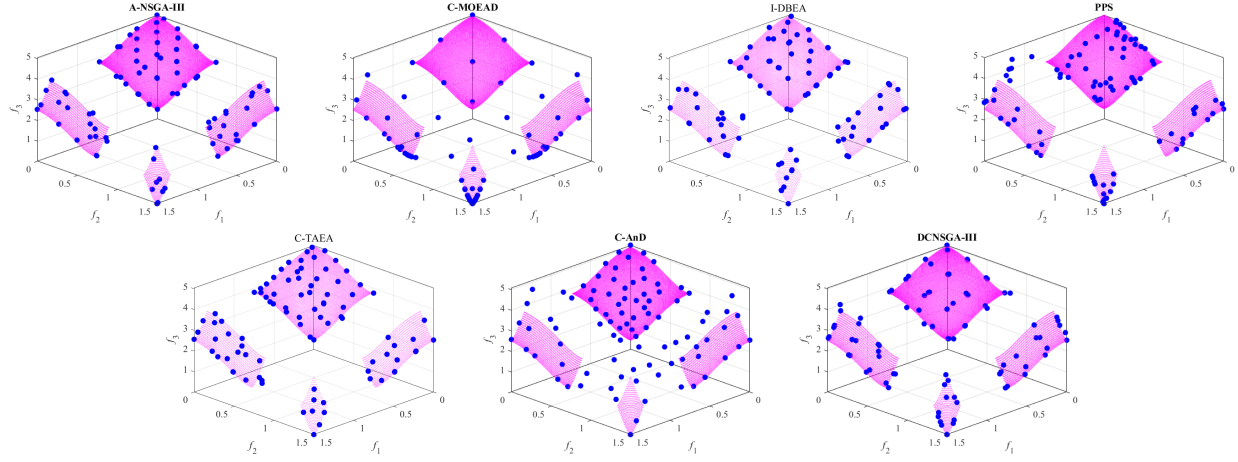


Fig. S-10. Obtained solutions for MW14 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

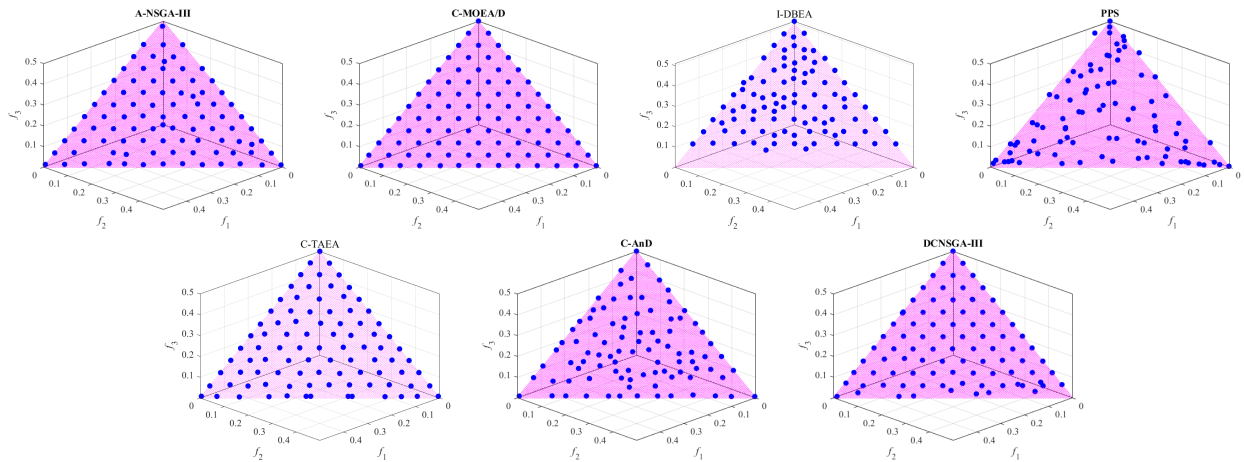


Fig. S-11. Obtained solutions for C1-DTLZ1 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

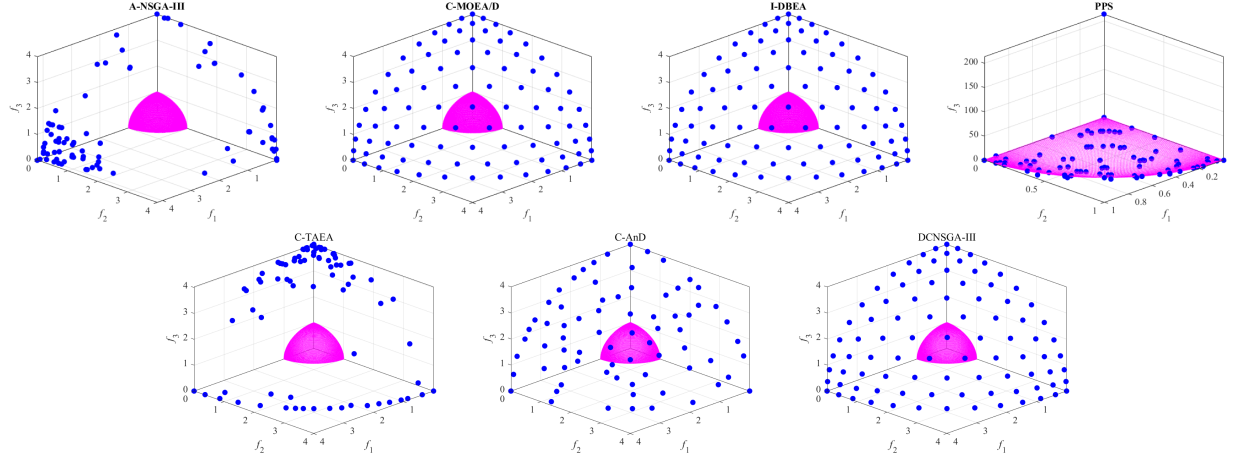


Fig. S-12. Obtained solutions for C1-DTLZ3 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

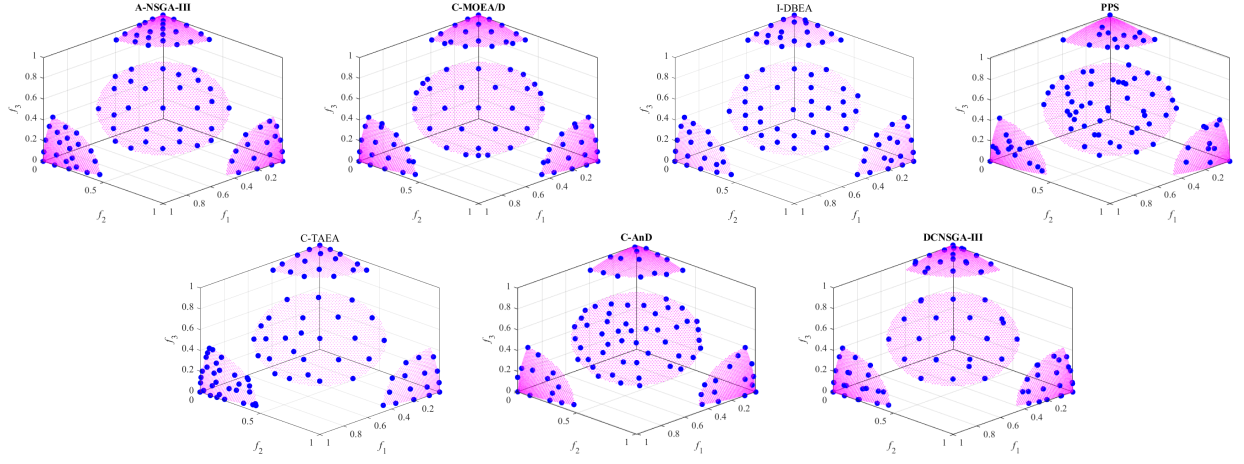


Fig. S-13. Obtained solutions for C2-DTLZ2 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

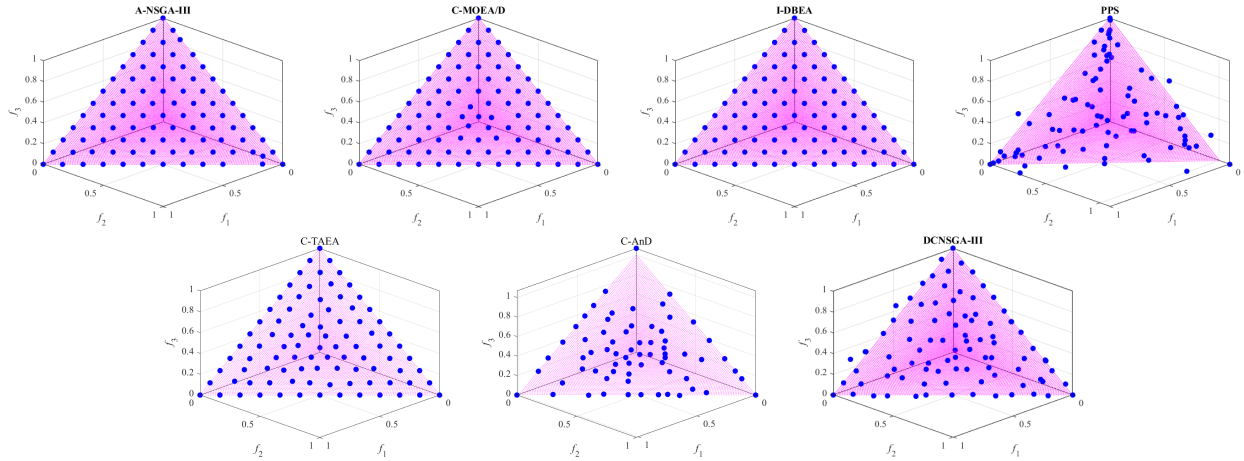


Fig. S-14. Obtained solutions for C3-DTLZ1 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

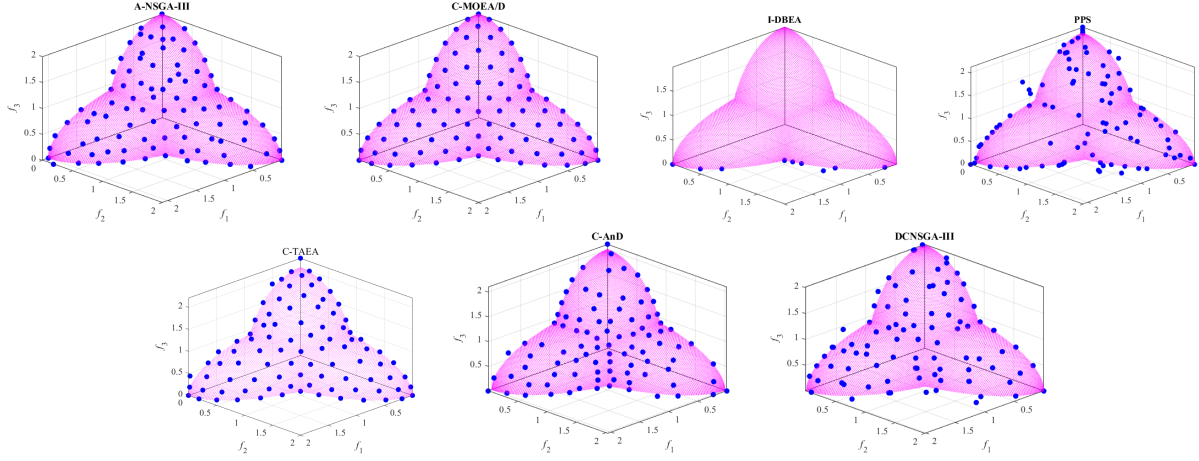


Fig. S-15. Obtained solutions for C3-DTLZ4 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

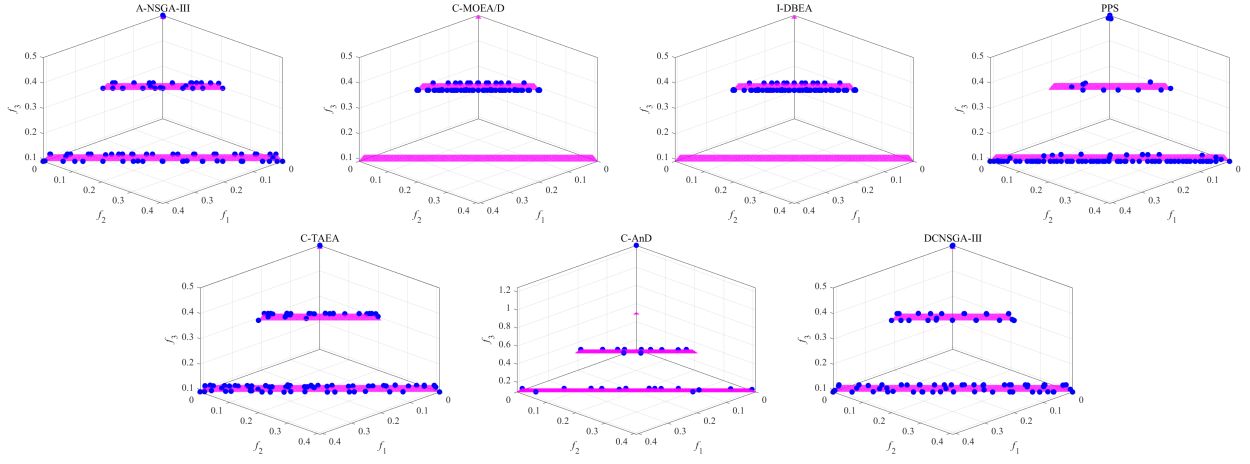


Fig. S-16. Obtained solutions for DC1-DTLZ1 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

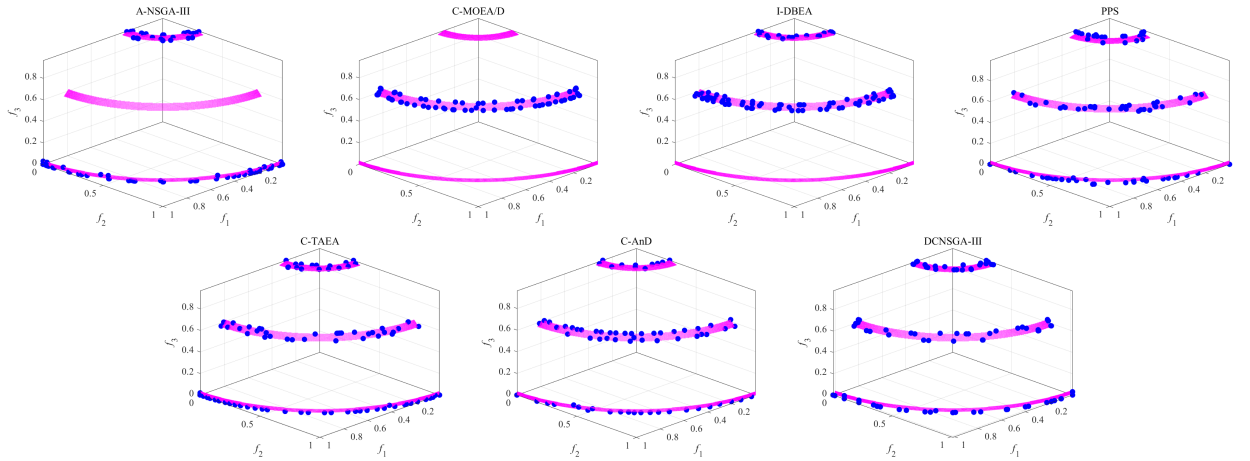


Fig. S-17. Obtained solutions for DC1-DTLZ3 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

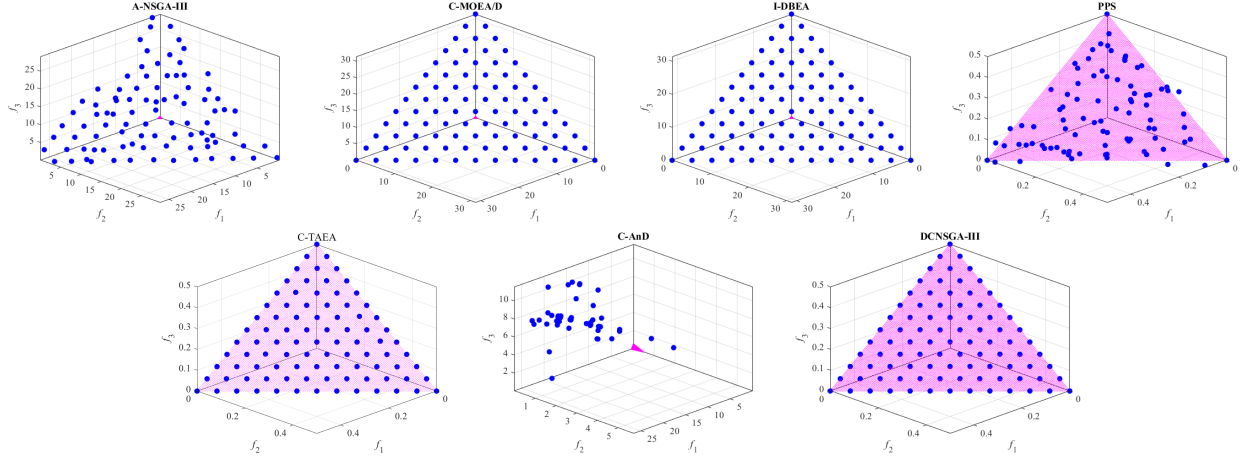


Fig. S-18. Obtained solutions for DC2-DTLZ1 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

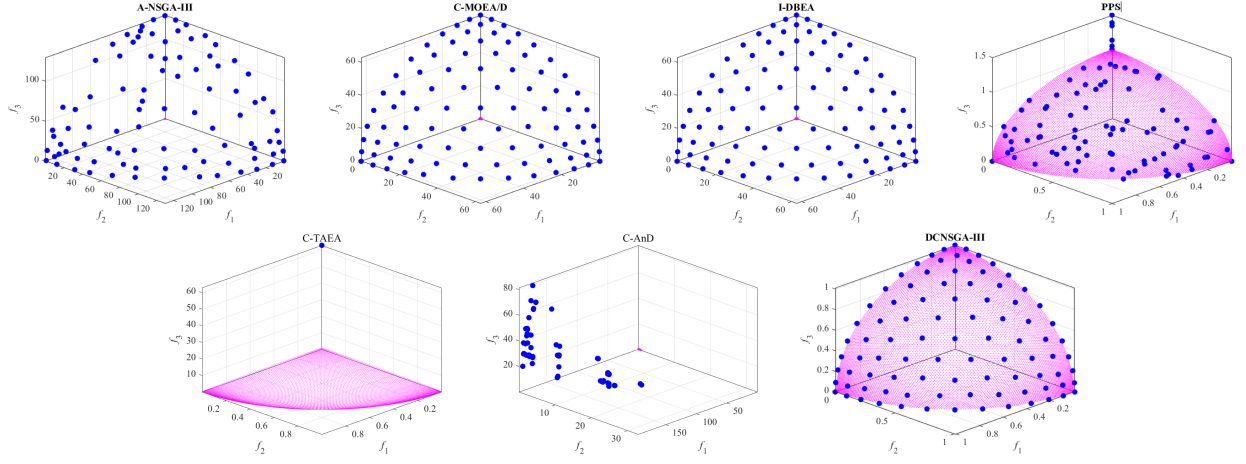


Fig. S-19. Obtained solutions for DC2-DTLZ3 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

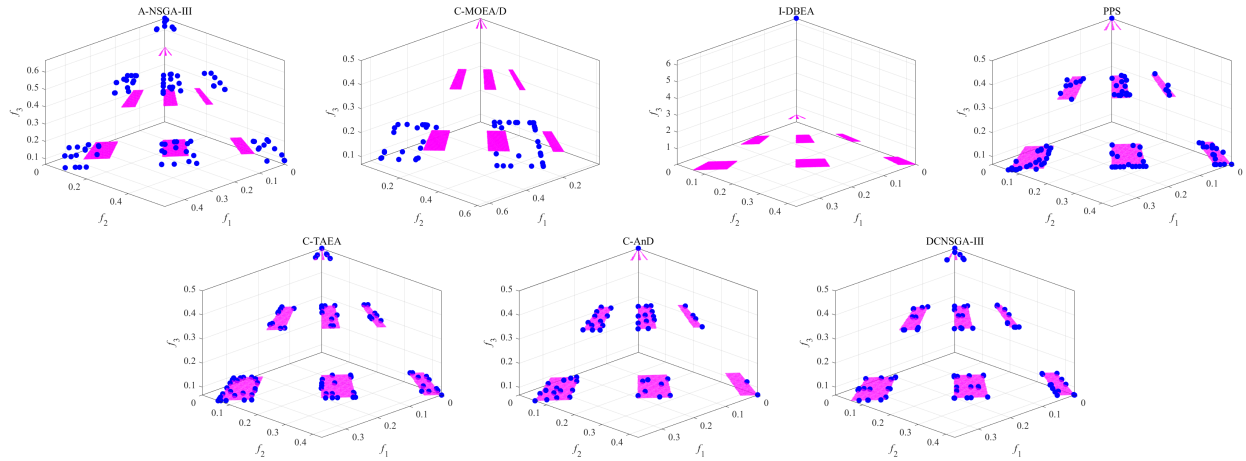


Fig. S-20. Obtained solutions for DC3-DTLZ1 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).

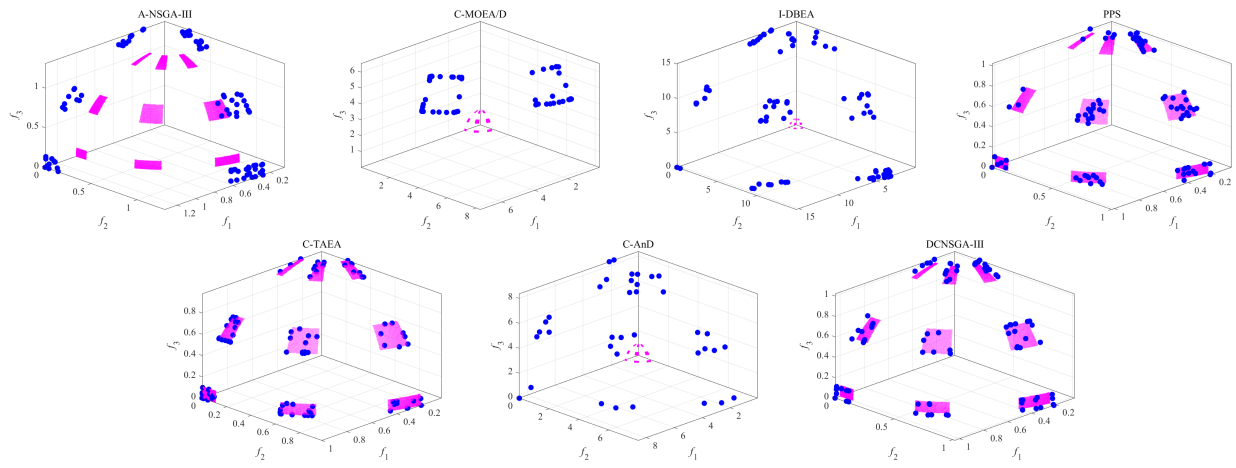


Fig. S-21. Obtained solutions for DC3-DTLZ3 problem using A-NSGA-III, C-MOEA/D, I-DBEA, PPS, C-TAEA, C-AnD, and DCNSGA-III from left to right (in median IGD value).